

UNIVERSITÀ DELLA CALABRIA

Dipartimento di Ingegneria Informatica,
Modellistica, Elettronica e Sistemistica

Dottorato di Ricerca in
Ingegneria dei Sistemi ed Informatica
XXV ciclo

Tesi di Dottorato

Supporting interaction and
cooperation in content-based web
applications

Marco Carnuccio

DIMES, Novembre 2013
Settore Scientifico Disciplinare: ING-INF/05

UNIVERSITÀ DELLA CALABRIA

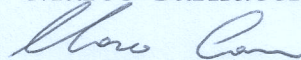
Dipartimento di Ingegneria Informatica,
Modellistica, Elettronica e Sistemistica

Dottorato di Ricerca in
Ingegneria dei Sistemi ed Informatica
XXV ciclo

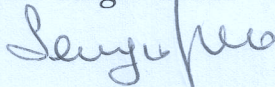
Tesi di Dottorato

Supporting interaction and cooperation in content-based web applications

Marco Carnuccio

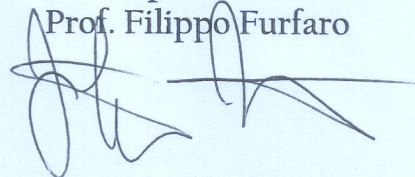


Coordinatore
Prof. Sergio Greco



Supervisore

Prof. Filippo Furfaro



DIMES

DIMES- DIPARTIMENTO DI INGEGNERIA INFORMATICA, MODELLISTICA,
ELETTRONICA E SISTEMISTICA
Novembre 2013

Settore Scientifico Disciplinare: ING-INF/05

To my family

Preface

The past two decades have been characterized by an exponential growth of web applications designed for a wide range of human areas, from scientific research to enterprise activities, from leisure to business. The development of these applications, in general, is not easy, especially when a high-level of customization is required; furthermore, if it is necessary to support advanced analysis features with final suggestions/forecasts, it becomes mandatory to integrate different tools.

In this thesis we present an innovative and integrated framework for web application development; the whole framework is composed by a powerful set of tools and offers a wide range of features: Document and Content Management, Social Cooperation, Knowledge Discovery and Business Process Management.

We start our treatment by introducing the core of the framework, named *Borè*. *Borè* is a novel semantic-based framework that realizes Web3.0 principles and it represents a tool for the next-generation Internet. This architectural paradigm allows high levels of customization to be reached and it facilitates and enriches user's web browsing experience. *Borè* is extremely innovative in two respects. Foremost, it allows Web information to be defined, organized, stored, queried and displayed as customizable objects and relations: an inexpert user can simply create the required Web. Secondly, *Borè* directly supports social networks (i.e., Social Cooperations), which spontaneously arise when users share resources among each others.

Rooted on these backgrounds, our work continues with two extensions of the core framework with the aim of building an innovative and complete tool for both Business Process Management and Knowledge Discovery. We first start by investigating and summarizing the state-of-the-art techniques for both topics, then we introduce innovative solutions for these purposes and we show advantages of the proposed techniques.

In particular we focus on recommender systems and on probabilistic approaches to recommendation. We present our extensions to classic probabilistic topic models, introducing causality and dependency through a sequential approach. Our contributions on this topic can be summarized in these directions. (i) we propose a unified probabilistic framework to model dependency in preference data, and instantiate the framework in accordance to different sequential assumptions; (ii) we study and ex-

perimentally compare the proposed models, highlight relative advantages and weaknesses; (iii) we adapted these models to support a recommendation scenario to generate personalized and context-aware recommendation lists; (iv) we show that the proposed sequential modeling of preference data better models the underlying data, as it allows more accurate recommendations.

Acknowledgements

During the development of this thesis, it has been my fortune and my privilege to work with brilliant professionals and, at the same time, wonderful people. My sincere gratitude goes to my PhD advisor Filippo Furfaro and to Luigi Pontieri for their continuous support and guidance both in professional and personal life. I hope to have always people like them by my side. A special thank goes to Giuseppe Manco for his help, motivation, knowledge and worthy ethical standards. I extend my gratitude to the whole research group I worked with during this years and those who believed in me. Last, but not least, I would like to thank the professor and friend Domenico Saccà, who made this amazing adventure possible.

Rende,
November 2013

Marco Carnuccio

Prefazione

Gli ultimi decenni sono stati caratterizzati da una crescita esponenziale delle applicazioni web create per settori differenti, dalla scienza al mondo aziendale, dallo svago al business. In generale, lo sviluppo di queste applicazioni non è semplice, specialmente quando è richiesto un alto livello di personalizzazione; inoltre, se è necessario supportare funzionalità avanzate di analisi allo scopo di fornire suggerimenti/previsioni, diventa obbligatorio integrare tool differenti.

In questa tesi viene presentato un framework per lo sviluppo di applicazioni web integrato ed innovativo; la piattaforma è composta da un potente insieme di strumenti ed offre un ampio ventaglio di funzionalità: Document and Content Management, Social Cooperation, Knowledge Discovery e Business Process Management.

Il lavoro di tesi si apre con l'introduzione del framework di base, chiamato *Borè*, che rappresenta una piattaforma innovativa di tipo semantico e si presenta come un tool per Internet della nuova generazione. Tale paradigma architetturale permette di raggiungere altissimi livelli di personalizzazione facilitando ed, al contempo, arricchendo l'esperienza di navigazione degli utenti finali. *Borè* è estremamente innovativo in due aspetti. Prima di tutto permette di definire, organizzare, salvare, interrogare e visualizzare le informazioni web sotto forma di nodi e relazioni personalizzabili: un utente non esperto è in grado di creare, in maniera semplice ed intuitiva, il web "desiderato". Inoltre *Borè* supporta direttamente il paradigma delle social network, che nascono spontaneamente nel momento in cui gli utenti condividono risorse.

Sulla base di questi concetti, il nostro lavoro continua con due estensioni del framework allo scopo di creare uno strumento completo ed innovativo con funzionalità di Business Process Mining e di Knowledge Discovery. In una prima fase viene investigato e descritto lo stato dell'arte relativo ad entrambi i topic, in seguito sono introdotte soluzioni innovative per le quali sono dimostrati i vantaggi competitivi.

Il nostro focus si è concentrato sui sistemi di raccomandazione ed, in particolare, sugli approcci probabilistici. Nello specifico vengono presentate delle estensioni ai modelli probabilistici classici, che introducono causalità e dipendenza tra le osservazioni, grazie ad un approccio sequenziale. I nostri contributi relativamente a questo topic possono essere così sintetizzati: (i) proposta un frameworwk proba-

bilistico unificato per modellare le dipendenze tra le osservazioni; tale framework viene istanziato seguendo differenti assunzioni sequenziali; (ii) studio e comparazione sperimentale dei modelli proposti, evidenziando relativi vantaggi e svantaggi; (iii) adozione di tali modelli nello scenario delle raccomandazioni al fine di generare liste di suggerimenti personalizzate e contestuali; (iv) dimostrazione che l'approccio sequenziale proposto modella in maniera più efficace i dati, in quanto permette di generare suggerimenti più accurati.

Ringraziamenti

Durante la preparazione di questa tesi, ho avuto la fortuna e l'onore di lavorare con professionisti brillanti e, al contempo, persone splendide. I miei più sentiti ringraziamenti vanno al mio tutor Filippo Furfaro e a Luigi Pontieri per il loro continuo supporto e la loro guida sia nell'ambito lavorativo che in quello personale. Mi auguro di poter avere sempre persone come loro al mio fianco. Un ringraziamento speciale va a Giuseppe Manco per il suo aiuto, la sua motivazione, la sua conoscenza e i suoi ammirevoli standard etici. Estendo la mia gratitudine a tutto il gruppo di ricerca con il quale ho collaborato in questi anni e a tutti coloro che hanno creduto in me. Infine desidero ringraziare il professore ed amico Domenico Saccà, grazie al quale ho vissuto questa splendida avventura.

Rende,
Novembre 2013

Marco Carnuccio

Contents

1	Introduction	1
1.1	Objectives	1
1.2	Publications	2
1.2.1	Journals	2
1.2.2	Book chapters	2
1.2.3	Papers in refereed conference proceedings	2
1.2.4	Other publications	3
1.3	Organization of the thesis	3
2	Borè: an architectural paradigm for content-based web applications ..	5
2.1	Introduction	5
2.2	Motivation and background	5
2.3	Framework	6
2.3.1	Preliminaries	6
2.3.2	Architecture	8
2.4	Data model	11
2.5	Graph querying	14
2.6	Social cooperation	17
2.6.1	Privileges management	17
2.7	Borè vs existing technologies	18
2.8	Case studies	19
2.8.1	A toy case study	19
2.8.2	A real-life case study	20
3	Caldera: enhancing Borè	27
3.1	Introduction	27
3.2	Advanced business process management	27
3.2.1	Workflow engine	30
3.2.2	Process discovery	32
3.2.3	Performance prediction	36
3.2.4	A case study: University of Calabria internships platform ...	41

3.3	Recommendation engine	44
3.3.1	Collaborative filtering approach to recommendation	47
4	Probabilistic topic models for sequence data	51
4.1	Introduction	51
4.2	Probabilistic topic models	51
4.3	Beyond the “bag-of-words” assumption	54
4.4	Modeling sequence data	55
4.4.1	Log-likelihoods	62
4.4.2	Estimating the hyper parameters	63
4.5	Application to recommender systems	63
4.6	Experimental evaluation	65
4.6.1	Perplexity	65
4.6.2	Recommendation accuracy	68
4.7	Other sequential approaches	75
5	Conclusion	81
	References	85

Introduction

1.1 Objectives

The past two decades have been characterized by an exponential growth of web applications designed for a wide range of human areas, from scientific research to enterprise activities, from leisure to business. The development of these applications, in general, is not easy, especially when a high-level of customization is required; furthermore, if it is necessary to support advanced analysis features with final suggestions/forecasts, it becomes mandatory to integrate different tools.

In this thesis we present an innovative and integrated framework for web application development; the whole framework is composed by a powerful set of tools and offers a wide range of features: Document and Content Management, Social Cooperation, Knowledge Discovery and Business Process Management.

The core of the framework, named *Borè*, is a novel semantic-based framework that realizes Web3.0 principles and it represents a tool for the next-generation Internet. This architectural paradigm allows high levels of customization to be reached and it facilitates and enriches user's web browsing experience. *Borè* is extremely innovative in two respects. Foremost, it allows Web information to be defined, organized, stored, queried and displayed as customizable objects and relations: an inexperienced user can simply create the required Web. Secondly, *Borè* directly supports social networks (i.e., Social Cooperations), which spontaneously arise when users share resources among each others.

The core architecture has been extended with innovative features with the aim of defining a complete and unified framework named *Caldera*, a platform for managing and analyzing collaborative process, which integrates an advanced recommender system.

These two topics have been widely investigated to accomplish this thesis and innovative solutions have been proposed for both areas:

- **Business Process Management.** We have extended classical process mining techniques with a predictive process-mining approach, which fully exploits context information, and determines the right level of abstraction on log traces in

data-driven way. Combining several data mining and data transformation methods, the approach allows the recognition of different context-dependent process variants, while equipping each of them with a separate regression model. Interesting results have been obtained on a real application scenario, showing that the proposed method is precise and robust.

- **Recommendation System.** We have focused on probabilistic topic models and we have extended the popular LDA model relaxing the bag-of-word assumption by hypothesizing that the current observation depends on previous information. Proposed models provide a better framework for modeling contextual information in a recommendation scenario, when the data exhibits intrinsic temporal dependency. These results have been confirmed by an experimental evaluation over real-life datasets.

1.2 Publications

The following publications have been produced while accomplishing this thesis.

1.2.1 Journals

- Barbieri N., Manco G., Ritacco E., Carnuccio M., Bevacqua A.: *Probabilistic Topic Models for Sequence Data*. Machine Learning Journal, Volume 93, Issue 1, pp. 5–29, October 2013.

1.2.2 Book chapters

- Bevacqua A., Carnuccio M., Folino F., Guarascio M., Pontieri L.: *A Data-Driven Prediction Framework for Analyzing and Monitoring Business Process Performances*. To appear in: Lecture Notes in Business Information Processing (LNBIP) — Springer-Verlag, 2013.

1.2.3 Papers in refereed conference proceedings

- Bevacqua A., Carnuccio M., Ortale R., Ritacco E.: *A new Architectural Paradigm for Content-based Web Applications: Borè*. Proceedings of the 15th International Database Engineering and Applications Symposium (IDEAS11). Lisbon, Portugal, September 2011.
- Bevacqua A., Carnuccio M., Cuzzocrea A., Ortale R., Ritacco E.: *Enforcing Interaction and Cooperation in Content-Based Web 3.0 Applications*. Web Technologies and Applications — 14th Asia-Pacific Web Conference (APWeb12). Kuming, China, April 2012.
- Barbieri N., Bevacqua A., Carnuccio M., Manco G., Ritacco E.: *Probabilistic Sequence Modeling for Recommender Systems*. Proceedings of the 4th International Conference on Knowledge Engineering and Knowledge Management (KDIR12). Barcelona, Spain, October 2012.

- Bevacqua A., Carnuccio M., Cuzzocrea A., Ortale R., Ritacco E.: *A Semantic-based Framework for Supporting Interaction and Cooperation in Content-Based Web 3.0 Applications*. Proceedings of the 21st Italian Symposium on Advanced Database Systems (SEBD13). Roccella Jonica, Italy, July 2013.
- Bevacqua A., Carnuccio M., Folino F., Guarascio M., Pontieri L.: *A Data-Adaptive Trace Abstraction Approach to the Prediction of Business Process Performances*. Proceedings of the 15th International Conference on Enterprise Information Systems (ICEIS13). Angers Loire Valley, France, July 2013. *Winner of the Best Paper Award*.
- Bevacqua A., Carnuccio M., Folino F., Guarascio M., Pontieri L.: *Adaptive Trace Abstraction Approach for Predicting Business Process Performances*. Proceedings of the 21st Italian Symposium on Advanced Database Systems (SEBD13). Roccella Jonica, Italy, July 2013.

1.2.4 Other publications

- Bevacqua A., Carnuccio M., Ferragina M., Liguori A., Ortale R., Ritacco E.: *Borè: un framework open source orientato al web 3.0*. Atti della V Conferenza Italiana sul Software Libero (CONFSL11). Milan, Italy, June 2011.

1.3 Organization of the thesis

The remainder of this thesis is organized as follows: Chapter 2 introduces the architecture and discusses the advantages of the *Borè* framework; Chapter 3 presents *Caldera*, an extension of the proposed architecture that includes advanced features of knowledge discovery and business process management; these two topics are widely investigated in this chapter. Chapter 4 reports innovative respects introduced in the framework, concerning probabilistic approaches to recommendation. In particular, this chapter focuses on new extensions to classic probabilistic topic models, introducing causality and dependency through a sequential approach. This provides a better framework for modeling contextual information in a recommendation scenario when the data exhibits intrinsic temporal dependency. Finally, Chapter 5 discusses conclusions and future work.

Borè: an architectural paradigm for content-based web applications

2.1 Introduction

The Web is an evolving system, which tries to adapt to the needs of users. The transition to Web2.0, and, currently, to Web3.0, are the expression of this trend: the goal is to focus on the leading role of the end user in Web browsing, which should be supported by adequate tools. In this chapter, we propose *Borè*, an architectural paradigm for developing content-based web applications based on cooperative interaction, whose foundations are based on the Web3.0 principles.

The proposed architecture is extremely innovative in two respects. The first one is the possibility of defining, organizing, storing, querying and displaying the information as customizable objects and relations: a not-expert user can create the Web that he may prefer. The second one is the realization of social networks, which spontaneously arise when users share resources among each others.

The chapter is structured as follows: Section 2.2 describes motivation and background of the proposed architecture. Section 2.3 introduces some preliminaries and, then, proceeds to cover the architecture of the *Borè* platform. Section 2.4 widely discusses the data model and its customizations, while Section 2.5 presents an overview of the formal language used as unified interaction protocol among all components of the *Borè* platform. Section 2.6 treats the establishment of social networks and privileges management in *Borè*. Section 2.7 provides an overview of some established competitors and highlights the main advantages introduced by *Borè*. Finally, in Section 2.8, we present some exemplificative case studies that exploit proposed architecture's features.

2.2 Motivation and background

Web-browsing models aim at reaching a challenging goal: to turn the Web environment into an intelligent and proactive setting, in which users play a key role. This is known as the *Semantic Web* and involves the development of advanced Web interfaces, capable to enrich with semantics both the supplied information and the users'

activities, with the ultimate goal of offering customized access and support to each individual user. Hitherto, efforts both in industry and academia towards the Semantic Web can be categorized as follows.

The Web1.0 is a model that saw the Web as a large container of information provided by various types of organizations. Users had a window on the Web mainly through their own homepage. From the structural point of view, available information was statically organized into *taxonomies*.

The Web2.0 model tried to overcome the static nature of the previous one by enabling some interactions with the end users. This has fostered the development and delivery of Web services as well as the growth of user communities, which play a key role in the enrichment of the Web information. Paradigmatic examples of Web2.0 applications include forums, blogs and social networks. Users share their experiences and provide an initial interpretation of semantic information through the tagging system: in this respect, the term *folksonomy* [71, 72] was coined in contrast to the taxonomy of Web1.0.

Nowadays, the not yet definitive Web3.0 model tries to further evolve the Web into the personal *universe* of each user, thus introducing the concept of *portable personal Web*, that follows from the widespread adoption of new technologies and devices. The idea is to generate adaptive systems, such as [21], which record and analyze users' activities, in order to define suitable user profiles, whose knowledge is in turn useful to anticipate their preferences, expectations and tastes; in other words a digital life stream database will be created. As a matter of fact, the single individual becomes the core of the Web, thus making the *folksonomy* evolves in the *me-onomy*.

In this chapter, we propose a novel architectural paradigm, referred to as *Borè*, for developing content-based Web applications based on Web3.0 principles. The basic idea is that a Web system should adapt to users' requirements, hence it must be able to evolve in order to integrate new data and new functionalities without the intervention of experts: users must be enabled to create their own Web as a "little universe". These universes should be able to interact with each other, share objects and communicate in order to build custom social networks, or social cooperations. As a matter of fact, *Borè* is extremely innovative in two respects. Foremost, it allows Web information to be defined, organized, stored, queried and displayed as customizable objects and relations: an inexperienced user can simply create the required Web. The second interesting feature is the possibility of directly supporting social networks/cooperations, which spontaneously arise through user resource sharing.

2.3 Framework

In this section we introduce some preliminary concepts as well as the architecture of the devised *Borè* platform.

2.3.1 Preliminaries

The intuition behind *Borè* is to exploit some principles of the object-oriented programming [68, 85] in the context of Web-application development. The logic model

of a Web application is viewed as a directed graph \mathcal{G} , whose nodes are the resources, while edges denotes the relations among pairs of such resources. By resource we mean all those entities that pertain to the Web application: a page, media contents, users, communities and so forth. A link between two resources (e.g., two Web pages) is a relation modeled as a direct edge from the former resource to the latter one.

By following the ideas of semantic Web systems, that exploit definition languages such as OWL¹, RDFS² and RDF³, Web information can be generalized by structuring raw data in high-level content objects.

More precisely, each resource (resp. relation) is a pair $\{type, instance\}$: *type* is the general definition (i.e., the schema), of the resource (resp. relation) and it is composed by a set of *fields*, whereas *instance* is the type instantiation. A field is an atomic information, whose base type can be any among integer, real, string and so forth. The single inheritance property [68, 85] holds for both resources and relations. A resource (resp. relation) that inherits from another resource (resp. relation) acquires all of information within the latter in addition to its own. The separation in types and instances and the inheritance property bring the significant advantages of object-oriented software development in the design and implementation of Web applications. We here mention module reuse, simplified code development, ease of extension and maintenance and compactness of shared information.

Formally, the prototype \mathcal{G} of a Web application in *Borè* can be modelled as a tuple $\mathcal{G} = \{R, E, RTT, ETT, RT, ET\}$, whose constituents are introduced next. R is the set of all resources corresponding to the nodes of \mathcal{G} . E is the set of all relations (i.e., the set of edges of \mathcal{G}). RTT and ETT are, respectively, the resource and the relation type taxonomies: resource and relation types are stored in two dedicated taxonomies enabling type inheritance. The root of RTT is the type *Object*, while the root of ETT is *Edge*. A possible implementation of both RTT and ETT will be exemplified in Section 2.4. RT and ET are two forests of taxonomies. Each taxonomy in RT (resp. ET) is an inheritance hierarchy among resource (resp. relation) instances. RTT , ETT , RT and ET are useful constituents, that allow a simple management of Web browsing in terms of navigation operators as it will be described in Section 2.5.

In our formalization, the interaction of an user with a Web application can be seen as a visit on the prototype \mathcal{G} : by clicking on a link within the current Web page, which corresponds to following a relation that departs from the resource being currently experienced, the user moves to another Web page (i.e., visits another resource of the graph model).

The strengths following from the adoption of the aforesaid logic model for Web applications deployed within *Borè* are manifold. First, a simplified resource management and querying. In particular, query processing in response to user interactions can take advantage of the solid results and foundations in the field of graph theory [16]. Second, the possibility to dynamically define new resources and types. Third, different Web applications can share the same schema (i.e., the set of all re-

¹ <http://w3.org/TR/owl2-overview>

² <http://w3.org/TR/rdf-schema>

³ <http://w3.org/RDF>

sources and relations types) and differ only in their respective instances. Fourth: compactness and cooperation. As a matter of fact, different Web applications may share some resources, thus avoiding redundancies. Moreover, resource sharing allows the generation of social cooperations and social networks among users.

To exemplify the foregoing concepts, we introduce in Fig. 2.1 the graph (or logic model) of a toy Web application. The resources pertaining to the Web application (i.e., an institute, two workers, a student and an event) are represented by nodes. Edges between nodes correspond to relations between resources. Actually, the graph in Fig. 2.1 depicts an institute, with two workers and a student, that publishes news and organizes events. The fields of individual nodes (resources) as well as their *ids* will be explained in Section 2.4.

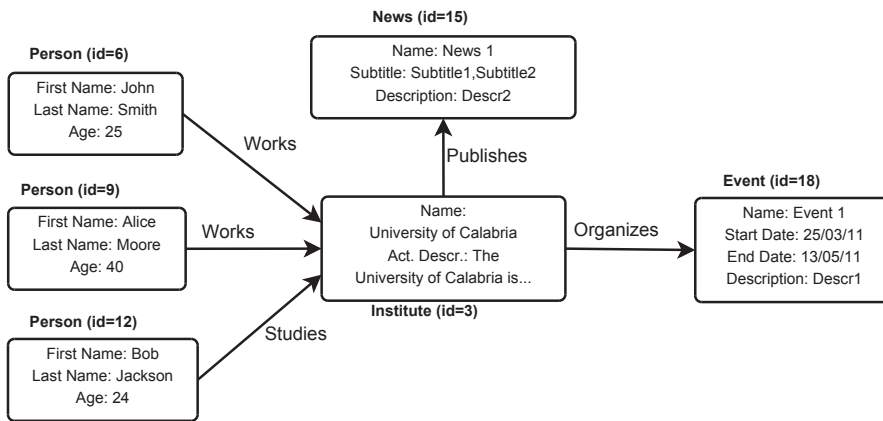


Fig. 2.1: The graph model of a toy web application

2.3.2 Architecture

The architecture of the *Borè* platform, shown in Fig. 2.2, is designed around the Model-View-Controller (MVC) design pattern [20]. This separates the three essential aspects of an application (i.e., business, presentation and control logic) with the purpose of considerably reducing both time and costs for development and maintenance. The architectural components of the *Borè* platform included within the Model, View and Controller layers are indicated in Fig. 2.2. In the following we point out main features of each layer.

View

The *View* is responsible both for the navigation of the web application graph and for query-result visualization. However, being in a Web environment, the final rendering of the information delivered by any Web application deployed within *Borè* is delegated to the user browser, which is the actual GUI. Query answers are delegated

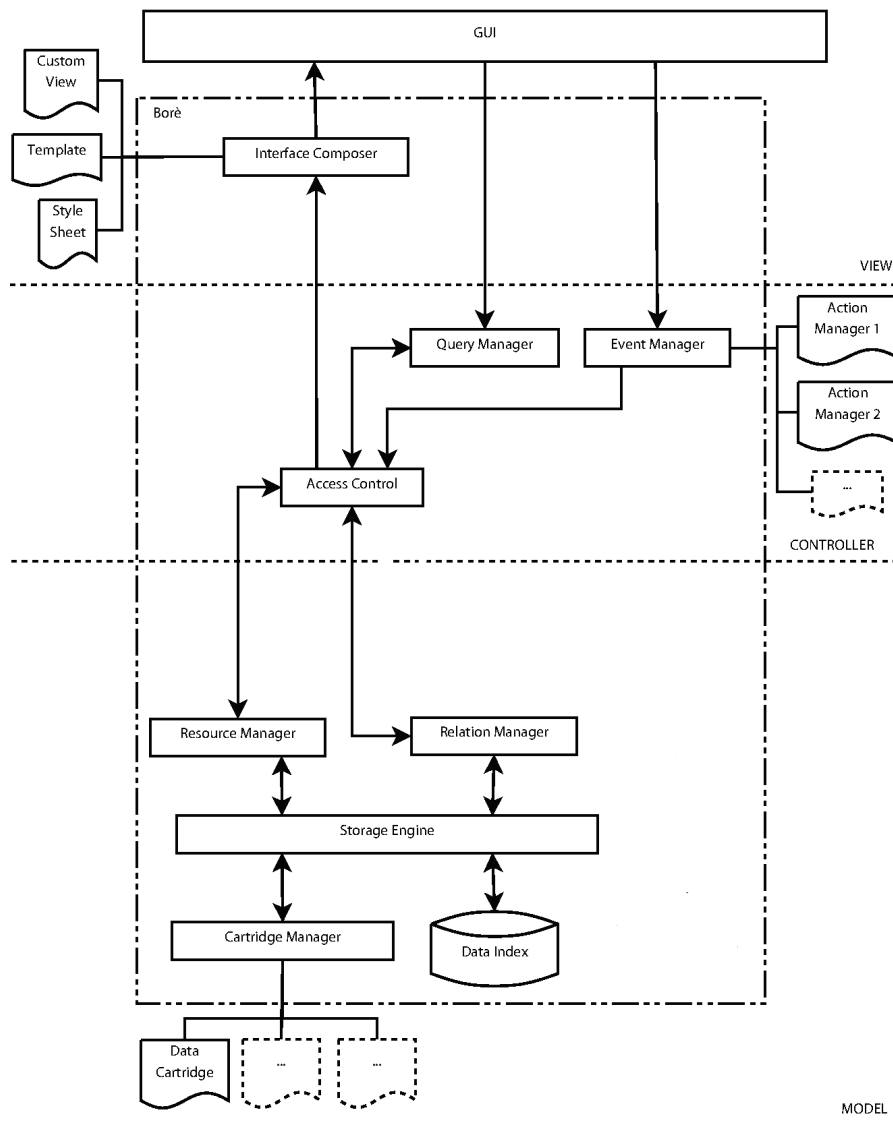


Fig. 2.2: Borè architecture

to the *Interface Composer* module, that can communicate with other architecture layers.

The Interface Composer relies on some pluggable, customizable and extensible modules, that can be categorized as follows:

- *Template*. It is associated to each resource type, and specifies the whole layout of the individual Web pages;
- *Style Sheet*. It specifies the rendering of the different resources appearing in a single Web page;
- *Custom View*. Useful with all those resources, whose rendering is not performed by the two preceding modules.

The View essentially decouples the presentation of the individual resources from the layout structure of the Web pages in which they appear.

More specifically, the user interface is built automatically from the node of the graph being currently visited. Such interface displays information concerning the current resource and some further resources that are related to the former. These latter resources are selected from those nodes of the graph of the Web application that are connected to the current node through relations. A view is provided by default: users are able to immediately browse resources and relations, in a plug-and-play environment.

Controller

The *Controller* manages information processing. It is composed by the following modules:

- *Query Manager*. Module that interprets user requests (i.e., interaction with the Web application) and translates them into the corresponding browsing or update operations on the graph;
- *Access Control*. It represents, essentially, a message dispatcher that coordinates nearly all modules of the *Borè* platform and controls the users' access to resources;
- Each resource can be associated to one or more events. The *Event Manager* module handles the events through pluggable, customizable and extensible *Action Managers*, which differ based on the nature of the events to notify to users. Some action managers should be provided by default (e.g., timers, email notifiers and so forth).

Model

The *Model* component is the data storage layer. It offers primitive functions both for query answering and for updating the graph of the Web application as well as the taxonomies of the relative resources and relations.

Resources and relations are separately managed by the *Resource Manager* and the *Relation Manager*, respectively. The Resource Manager (resp. Relation Manager)

is responsible for accessing, storing, updating and versioning the individual resources (resp. relation) and the resource (resp. relation) taxonomies. Hence, Resource and Relation Managers are a representation of the Web application graph.

Resource and Relation Managers communicate with the *Storage Engine*, which handles raw data. Raw data is retrieved through the *Data Index*, which contains meta data information, and the *Cartridge Manager*, which loads, stores and updates the various instances by means of pluggable, customizable and extensible storage units (e.g., relational DBs, files, and so forth).

2.4 Data model

Borè neatly separates the schema of the data from its instances through the Model component. In this section we provide an explanation of the data model, showing how raw data are stored and managed within *Borè*. Both the resource type taxonomy *RTT* and the relation type taxonomy *ETT* introduced in Section 2.3 are stored within the Data Index. A possible implementation of the Data Index in the context of the example of Fig. 2.1 is shown in tables 2.1, 2.2 and 2.3.

Resource Types		
id	name	father_id
1	Object	null
2	Node	1
3	Community	1
4	News	2
5	Event	2
6	Person	3
7	Institute	3

Relation Types		
id	name	father_id
1	Edge	<i>null</i>
2	Publishes	1
3	Collaborates	1
4	Organizes	2
5	Works	3
6	Studies	3

Table 2.1: Resource type table

Table 2.2: Relation type table

Type Fields				
id	type_id	name	type	Cardinality
1	2	title	string	1.1
2	2	description	text	0.1
3	4	subtitle	string	0.n
4	5	start date	date	1.1
5	5	end date	date	0.1
6	6	first name	string	1.1
7	6	last name	string	1.1
8	6	age	int	1.1
9	7	name	string	1.1
10	7	activity description	string	0.1

Table 2.3: Resource field table

Resource Instances		
id	type_id	father_id
1	1	<i>null</i>
2	3	1
3	7	2
4	1	<i>null</i>
5	3	4
6	6	5
7	1	<i>null</i>
8	3	7
9	6	8
10	1	<i>null</i>
11	3	10
12	6	11
13	1	<i>null</i>
14	2	13
15	4	14
16	1	<i>null</i>
17	2	16
18	5	17

Table 2.4: Resource instance table

Relation Instances			
id	type_id	source	dest
1	5	6	3
2	5	9	3
3	6	12	3
4	2	3	15
5	4	3	18

Table 2.5: Relation instance table

Int Assignments				
id	instance_id	field_id	value	pos
1	6	8	25	0
2	9	8	40	0
3	12	8	24	0

Table 2.6: Int assignment table

String Assignments				
id	instance_id	field_id	value	pos
1	3	9	Un. of Calabria	0
2	3	10	The Un. of Calabria is	0
3	6	6	John	0
4	6	7	Smith	0
5	9	6	Alice	0
6	9	7	Moore	0
7	12	6	Bob	0
8	12	7	Jackson	0
9	14	1	News1	0
10	15	3	Subtitle1	0
11	15	3	Subtitle2	1
12	17	1	Event1	0

Table 2.7: String assignment table

Text Assignments				
id	instance_id	field_id	value	pos
1	14	2	Descr1	0
2	17	2	Descr2	0

Table 2.8: Text assignment table

Date Assignments				
id	instance_id	field_id	value	pos
1	18	4	25/03/11	0
2	18	5	13/05/11	0

Table 2.9: Date assignment table

The *Resource Type* table contains the type definitions and stores the taxonomy. A resource is identified by an unique *id*, a *name* and a resource type parent, *father_id* (single inheritance property); note that the root has no parent. In the proposed schema, there are seven resource types: Object, Node, Community, News, Event, Person and Institute, whose taxonomy is depicted in Fig. 2.3. The *Resource Field* table contains the information about the fields of a type: each field is related to a single resource and it has its own base-type and cardinality.

To better elucidate, let us consider the type with *id*5, whose name is *Event*. Its taxonomy path is *Object* (*id*1) → *Node* (*id*2) → *Event* (*id*5). The specific fields of *Event* are *start date* and *end data*. However, since *Event* inherits from *Node*, it also borrows the fields of the latter (i.e., *title* and *description*).

The definition of the *Relation Type* table is analogous to the one of the *Resource Type* table: there is an *id*, a *name* and the relation parent id (*father_id*). In the example of Fig. 2.1, there are six types of relations, namely, Edge, Publishes, Collaborates, Organizes, Works and Studies, whose taxonomy is illustrated in Fig. 2.4.

The separation of data from its schema in *Borè* allows a new custom type to be easily defined, which trivially involves to add new tuples in *Resource Type*, *Relation Type* and *Resource Field* tables.

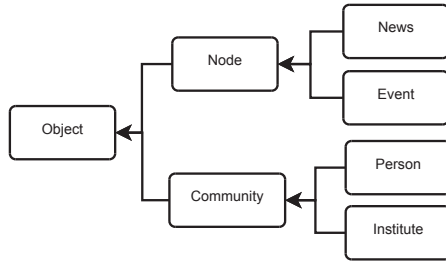


Fig. 2.3: Resource type taxonomy

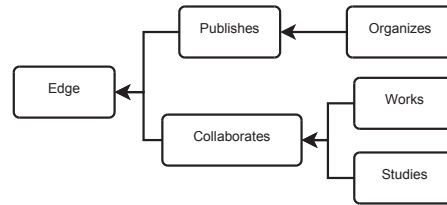


Fig. 2.4: Relation type taxonomy

The availability of a Data Index, that defines both the resource-type and the relation-type taxonomies allows a web application to be populated by storing the actual raw data as instances within suitable cartridges. These can be heterogeneous external data sources, viewed as additional modules in the context of the architecture of the *Borè* platform. Possible examples of exploitable cartridges include relational databases, file systems, XML databases, map-reduce storage units, remote contents and so forth. A possible cartridge implementation for the context of the example in Fig. 2.1 is provided by the relational database shown in tables 2.4, 2.5, 2.6, 2.7, 2.8 and 2.9.

The *Resource Instance* table stores the instances of the Web application. By looking at the attribute *type_id*, one can recognize all the resource types that compose the graph of Fig. 2.1, namely, an *Institute* (*id* 3), three *Persons* (two workers with *id* 6 and 9 and one student with *id* 12), one news (*id* 15) and one event (*id* 18). For the sake of clearness Fig. 2.1 has been simplified in expressing taxonomies.

Taxonomies are represented through the attribute *father_id*. To exemplify, the person id 12 inherits from the resource id 11 (which is a *community* instance) which, in turn, inherits from id 10, an *Object* instance. Instances' fields are mapped in several tables, one for each base-type. The field tables for the example of Fig. 2.1 are *String*, *Int*, *Date* and *Text Assignments*. Let's consider the person with id 12. In the string assignment table there are two entries associated with that person (i.e., the ones with id 7 and 8). The former has *value* equals to *Bob* and its *field_id* is 6; this is an external key for the table *Resource Fields* (see tab. 2.3): this means that the name of the field is *first name* and its cardinality is 1.1. As a matter of fact, the first (and unique) name of person id 12 is *Bob*. The analysis of field id 8 reveals that the *last name* of the person with id 12 is *Jackson*. Notice that the *pos* attribute is necessary to distinguish between different values for a field with cardinality greater than 1: logically, a field with more than one value, is an array indexed by the attribute *pos*.

The *Relation Instance* table stores links between pairs of resources. Its schema includes a key attribute *id*, an external key (*type_id*) for the relation type table (see tab. 2.2) and two further external keys for the resource instance table, namely *source* and *dest*, that represent the source and the destination of the corresponding directed edge in the graph.

Consider the relation id 5. Its *type_id* is 4, which means that it is an *Organizes* relation. The relation source is resource id 3 (whose type is *Institute*) and the relation destination is resource id 18 (an *Event*). An illustration of this relation with related resource taxonomies is shown in Fig. 2.5. The picture shows blocks with rounded corners that correspond to instances. These blocks are connected with irregular and dotted blocks, that provide details on the corresponding instances (i.e., their types and fields). Block containment denotes inheritance. Precisely, *University of Calabria* (id3) is an *Institute*, which inherits from a *Community* (id2) as well as from an *Object* (id1) (since the *Community* with id2 in turn inherits from the *Object* with id1). *University of Calabria* has two fields, namely (*Name* and *Activity Description*), and is linked to an *Event* (id18), through the relation *Organizes*. *Event*, in turn, inherits from a *Node* (id17) and from an *Object* (id16). *Event's* fields are *StartDate*, *EndDate*, *Name* and *Description*. Notice that *Name* and *Description* are inherited from the *Node* instance (id17).

2.5 Graph querying

The Query Manager module translates each user interaction — simple (i.e., click stream) and advanced queries (i.e., performed via a specific form) — into a suitable mathematical set language, that is well suited to graph-based interpretation of the Web application itself. This language is the actual engine of the *Borè* platform and allows a high degree of modularity: each architectural module can be modified, reused or re-implemented without modifying other components. In this section we review some fundamental aspects of such language, by exploiting the definitions expressed in Section 2.3.

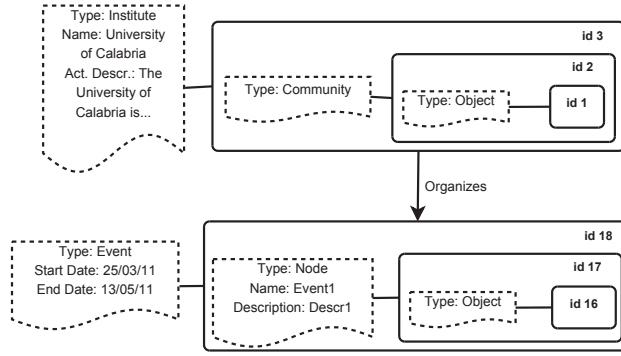


Fig. 2.5: An insight into the portion of the web graph of Fig. 2.1 including the resources *University of Calabria* and *Event1* and their relationships

The starting point is represented by some of its most basic operators. One operator is *IsA*: given a type t and a taxonomy T , such that $t \in T$, *IsA* returns all the types which lie on the path from the taxonomy root to t . Formally:

$$IsA(t, T) = \{t\} \cup \{t' | \exists t'' (t' \rightarrow t) \in T\} \cup \{t' | \exists t'' : (t'' \rightarrow t) \in T \wedge t' \in IsA(t'', T)\}$$

where $(x \rightarrow y)$ is a tree branch in which x is the parent, while y is the child; in other words y inherits from x .

Another operator is *type*: given a resource (relation) instance i and a type taxonomy T , *type* returns t , the last type (i.e., top-down following the hierarchy in T) i belongs to.

$$t = type(i, T)$$

The *typeList* operator is closely related to *IsA* and *type* ones: given a resource (relation) instance i and RTT , *typeList* is defined as follows:

$$typeList(i, RTT) = IsA(type(i, RTT), RTT)$$

By building on the three basic operators discussed above, it is possible to define more complex operators. For instance, *filterDown* and *filterUp* operators enable user browsing in the Web application. Given a resource r , *filterDown* permits to select all neighbor resources $\{r_1, \dots, r_n\}$ such that a direct edge from r to r_i (where $i = 1, \dots, n$) exists in the Web application graph.

Formally, given a resource r , two subsets $RTT' \subseteq RTT$ and $ETT' \subseteq ETT$, and a boolean function δ (that represents some generic binary property):

$$\begin{aligned} filterDown(r, RTT', ETT', \delta, RTT, ETT, R, E) = \{r' | r, r' \in R \wedge (r \Rightarrow r') \in E \\ \wedge IsA((r \Rightarrow r'), ETT) \cap ETT' \neq \emptyset \\ \wedge typeList(r', RTT) \cap RTT' \neq \emptyset \wedge r' \vdash \delta\} \end{aligned}$$

where notation $(x \Rightarrow y)$ indicates a relation from the resource x to the resource y and $r' \vdash \delta$ means that δ holds true on r' . Symmetrically, *filterUp* allows the navigation of indirect edges in the graph:

$$\begin{aligned} filterUp(r, RTT', ETT', \delta, RTT, ETT, R, E) = \{r' | r, r' \in R \wedge (r' \Rightarrow r) \in E \\ \wedge Isa((r' \Rightarrow r), ETT) \cap ETT' \neq \emptyset \\ \wedge typeList(r', RTT) \cap RTT' \neq \emptyset \wedge r' \vdash \delta\} \end{aligned}$$

To better understand the translation of user interactions with the Web front-end we next enumerate some example queries. Queries' results provide the new content of the Web front-end to supply to the end user in response to the aforesaid interactions. These queries are meant for the example of Fig. 2.1 and their role will be clarified in the case studies of Section 2.8.

$$\begin{aligned} filterDown (Un. of Calabria, \{Object\}, \{Publishes\}, \\ null, RTT, ETT, R, E) = \{News 1, Event 1\} \end{aligned} \quad (2.1)$$

This query returns all resources, whose types list contains *Object*, that are reachable from the *University of Calabria* node through a *Publishes* relation, assuming the taxonomies *RTT*, *ETT*, *R* and *E* and assuming there is no constraints (i.e., δ function is null).

$$\begin{aligned} filterDown (Un. of Calabria, \{News\}, \{Edge\}, \\ null, RTT, ETT, R, E) = \{News 1\} \end{aligned} \quad (2.2)$$

This query returns all resources, whose types list contains *News*, that are reachable from the *University of Calabria* node through an *Edge* relation, assuming the taxonomies *RTT*, *ETT*, *R* and *E* and assuming there is no constraints.

$$\begin{aligned} filterUp (Un. of Calabria, \{Person\}, \{Collaborates\}, null, \\ RTT, ETT, R, E) = \{Alice Moore, John Smith, Bob Jackson\} \end{aligned} \quad (2.3)$$

This query returns all resources, whose types list contains *Person*, that reach the *University of Calabria* node through a *Collaborates* relation, assuming the taxonomies *RTT*, *ETT*, *R* and *E* and assuming there is no constraints.

$$\begin{aligned} filterUp (Un. of Calabria, \{Person\}, \{Collaborates\}, \\ age < 30, RTT, ETT, R, E) = \{John Smith, Bob Jackson\} \end{aligned} \quad (2.4)$$

This query returns all resources, whose types list contains *Person*, that reach the *University of Calabria* node through a *Collaborates* relation, assuming the taxonomies *RTT*, *ETT*, *R* and *E* and assuming that we are looking for under 30 years old people.

$$\begin{aligned} filterUp (Un. of Calabria, \{Person\}, \{Works\}, \\ age < 30, RTT, ETT, R, E) = \{John Smith\} \end{aligned} \quad (2.5)$$

This query returns all resources, whose types list contains *Person*, that reach the *University of Calabria* node through a *Works* relation, assuming the taxonomies *RTT*, *ETT*, *R* and *E* and assuming that we are looking for under 30 years old people.

2.6 Social cooperation

In this section we discuss the second strong point of the proposed framework: the generation of a social cooperation environment.

One of the basic ideas of Borè is that the user should be free to create its own Web “universe” and share it with others. Even the user is a Web resource and, particularly, he represents the centre of his web.

Let us suppose that an user creates several Web applications and publishes them to other users. If such applications share resources, an environment of social cooperation and social networking spontaneously arises, since different users interact with one another.

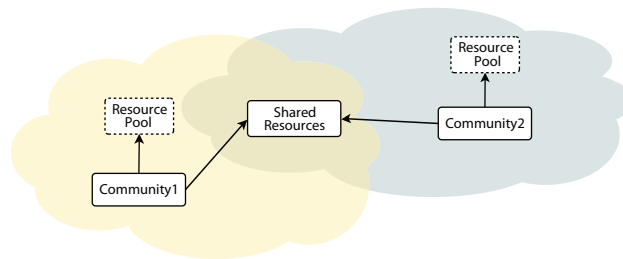


Fig. 2.6: Social cooperations

Fig. 2.6 shows a simple scenario with two communities. Each community has a set of owned resources, the *Resource Pool*) and some *Shared Resources*. Both the communities can read and, possibly, modify shared resources and can be notified on any interaction/update.

To elucidate, in the context of the toy example of Fig. 2.1, the two communities can be *University of Calabria* and *Alice Moore*. Hypothetically, we can imagine that the only shared resource between the two communities is *News 1*. This essentially enables a social cooperation between *University of Calabria* and *Alice Moore*; a detailed explanation is given in Section 2.8.

2.6.1 Privileges management

The social nature of the proposed architecture calls for a flexible but, at the same time, strong privileges management on the resources. In *Borè* we propose a Unix-like grant system. The difference is that, normally, each item is assigned to an owner and a group whilst, in our scenario, a node could be assigned to several communities; for this reason we choose to associate a rich set of privileges to each node on the graph.

A user, for instance, can access or edit a specific node in the graph because he is its owner or because he is a member of the community which created the resource.

Formally, a privilege G is a tuple $G = \{\alpha, \beta, \gamma\}$, where: α is the community the resource belongs to; β is a constraint over α . G can be assigned to the sub-set of α ,

for which β holds; γ is the privilege type (i.e., it is the set of all actions allowed on the resource).

Each resource can be associated to 0, 1 or more privileges, which specify available actions on the resource assigned to a subset of users. This is a very fine-grained privilege management (i.e., on each resource). In many practical cases, however, this approach might be too complex and impractical, especially in configurations with a large number of nodes. For this reason a coarse-grained configuration is appreciated. With this configuration it is possible to select and assign privileges for a group of resources with the same type. For example, a company might decide to publish (i.e., make visible to guests) all the news, but to reserve other types (e.g., invoices, personal information and so forth) only for the members of specific communities and/or their owners. This approach significantly simplifies privileges management, but at the same time decreases assignment flexibility. A mixed approach is, in our opinion, the optimal strategy. A set of coarse-grained privileges can be assigned and, when it is necessary, it is possible to define fine-grained privileges. By turning to the previous example, the company could deny to any guest the invoice querying (coarse-grained constraint), but it can give to an user the permission to see only his own invoices (fine-grained privileges).

2.7 Borè vs existing technologies

In this section, we discuss the main differences between *Borè* and some established competitors. Such competitors can be divided into two main categories, namely, frameworks and applications.

Frameworks include Web infrastructures such as Django⁴, Ruby on Rails⁵ and Symfony⁶. These are meant to support expert users in the development of applications based on the Model-View-Controller pattern. On the other hand, applications are content-management systems, such as Joomla!⁷, Drupal⁸ and Alfresco⁹, whose goal is to enable, even inexperienced users, to simply publish their contents on the Web through a wide variety of tools, components and templates. *Borè* exhibit meaningful differences w.r.t. both frameworks and applications.

In particular, one general limitation of frameworks is that they are not easily customizable and extensible. As a matter of fact, the definition of new structured types is usually a complex process that necessarily involves expert developers (e.g. in Django this involves a non trivial coding process). Moreover, the graphical tools for resource instantiation and manipulation will be accessible only to system administrators. Additionally, apart from back-office features, there is no ready to use representation

⁴ <http://djangoproject.com>

⁵ <http://rubyonrails.org>

⁶ <http://symfony-project.org>

⁷ <http://joomla.org>

⁸ <http://drupal.org>

⁹ <http://alfresco.com>

of the newly defined objects for the end user. Conversely, *Borè* allows the definition of new resources and relations through user-friendly embedded graphical tools (see Section 2.8), and it provides default representations of new instances. This suitable interface for type instantiation and instance manipulation offered, can be simply accessed by system administrators and user as well. *Borè* is also easily extensible through the implementation of well-defined interfaces (see Section 2.4).

Applications mainly provide the user with friendly tools for template definition. Unfortunately, apart from some exceptions (e.g., Drupal), such competitors generally do not allow the definition of new types and they lack of features for resource and relation manipulation in terms of taxonomies, that are instead provided by *Borè*. Therein, we recall that the latter is equipped with suitable user-friendly graphical tools for taxonomy management presented in Section 2.4 that will be exploited in Section 2.8. Additionally, applications do not support relation customization and manipulation, that are instead a strong point of *Borè*.

To summarize, *Borè* is a platform whose features not only comprise the characteristics of both frameworks and applications, but also include new and more advanced functionalities, such as the management of user and community-interactions through social cooperations and resource sharing.

2.8 Case studies

In this section we present two case studies; the first one is a toy case study aimed at highlighting some major features of *Borè*, the second one is a real-life case study that exploits *Borè* features in a complex scenario.

2.8.1 A toy case study

Case study presented in this section consists in the explanation of the toy Web application introduced in the example of Fig. 2.1. Let's suppose that the prof. *Alice Moore*, one of the workers of the institute *University of Calabria*, would like to call a Faculty Council.

The first step is the accessing the home page of the institute (see Fig. 2.7(a)): the *Borè* automatic output is essentially a collection of frames, each of which corresponds to a specific representation of Web contents. The main frame shows the description of a resource, whose type is *Institute* and whose realization is *University of Calabria*. The description, basically, contains the field values (i.e., *Name* and *Activity Description*). The side boxes contain the results of the operators described in Section 2.5 (i.e., *filterUp* and *filterDown*). In particular, the top frame (i.e., *Forward Link Box*) shows the result of the query 2.1 reported in Section 2.5, while the bottom one (i.e., *Backward Link*) reports the answer to the query 2.3.

When *Alice* follows the *Alice Moore* link, her personal home page is displayed (see Fig. 2.7(b)). The latter is again composed of a main frame that displays her personal information and of a set of side boxes. The *filterUp* and *filterDown* queries

to construct these side boxes are automatically issued when *Alice* moves from the *University of Calabria* page (Fig. 2.7(a)) to her personal Web page (Fig. 2.7(b)). Notice that, in the graph model of the toy example of Fig. 2.1, the resource *Alice Moore* has no incoming edges: hence, the *filterUp* operator returns an empty set and the backward link box is not displayed at all on her personal page. Additionally, the box named *Shared* contains all resources that *Alice Moore* shares in the Web application with *University of Calabria*: the *Shared* box actually connects *Alice Moore* to her social network.

Since *Alice* wants to call a *Faculty Council*, she need to create a new custom resource type and, then, to instantiate it. A form for the definition of a new resource type is shown in Fig. 2.7(c). On the left side, the tree structure shows the resource type taxonomy: once selected the father node, the new type will be appended in its hierarchy. On the right side there are tools for the definition of the new type fields.

The form for resource instantiation is depicted in Fig. 2.7(d). Such a form is used by *Alice* to create a resource (i.e., an instance) of the type *Faculty Council* by providing values for each field. *Alice* can also specify a connection between her personal page and the new resource.

Among other operations, *Alice* can send an invitation to involved people or associate a reminder that will dispatch emails to the participants before the Faculty Council. In *Borè*, these operations correspond, respectively, to the definition of a new relation (e.g., *Participates*), connecting the invited people and the council, and to the association of a custom *Action Manager* to the newly created resource *Faculty Council*.

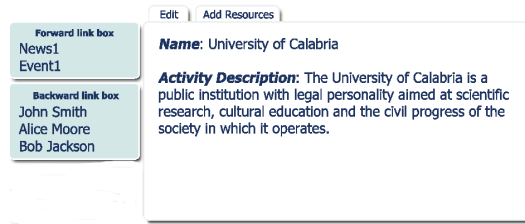
2.8.2 A real-life case study

Previous section presented a toy example with the aim of describing *Borè* features. An open source implementation of the proposed framework can be found online¹⁰. This software has been used for the realization of different web portals; see for example:

- <http://icar.cnr.it>, the web portal of the Institute for high performance computing and networking, a computer science research institute of Italian National Research Council (CNR);
- <http://openknowtech.it>, the portal of the OpenKnowTech project, a public-private laboratory which aims at developing and assess the adoption of open source for software development and use;
- <http://tetris.deis.unical.it>, the platform for the training related to the *TETRis* (Tetra Innovative Open Source Services) research project.

In this section we are going to analyze the Institute for high performance computing and networking (ICAR-CNR) web portal. This Institute is located at Rende (Cosenza) and has two branches in Naples and in Palermo. ICAR presently enrolls

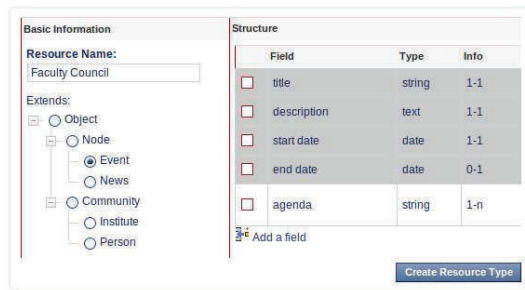
¹⁰ <http://oktlab.openknowtech.it/OKTLAB/it/newsview.wp?contentId=NEW425>



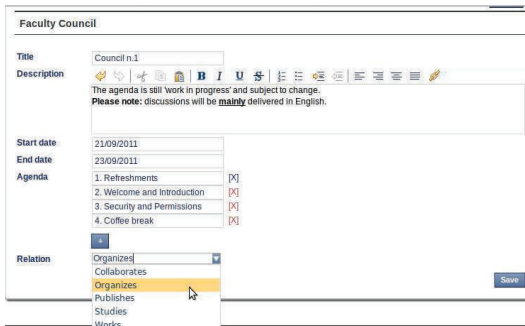
(a) Home page



(b) Personal page



(c) New type creation page



(d) Instance definition page

Fig. 2.7: Browsing, definition and instantiation of a new resource in Bore

over 100 employees (researchers, technicians and administrative staff), equally distributed on the three sites. In addition many professors from University of Calabria, University “Magna Graecia” of Catanzaro, University “Mediterranea” of Reggio Calabria, University “Federico II” of Naples and University of Palermo have research appointments with ICAR and actively cooperate in the research activities of the Institute.

The mission of the institute is to carry out research activities in the field of Computer Science and Information Technology, cooperating with the academic world and with other private and public research organizations. Some of the prominent research topics the Institute is engaged in are: Database and Knowledge Base Systems, Models and Tools for Parallel Processing, Representation and Control of Complex Systems, Knowledge Discovery in Databases and Data Mining, Artificial Intelligence, Multimedia Systems, Mathematical Computation, Artificial Vision.

In addition to its research activities, the institute is interested in the pre-competitive development and technological transfer of research results, and carries out educational and training activities, through scholarships and research fellowships, advanced after-university specialization courses, and non-university higher education activities. The scientific and administrative responsibility of the Institute is assigned to a Director, while for each section there is a Responsible. The management roles take advantage of the support of the following committees: the Scientific Council, the Institute Committee, and the Section Committees (one for each section).

People at ICAR-CNR may work either in the administrative or in the research area in several different positions. Furthermore, scholarship students, fellowship students and part-time people collaborate with the institute staff. Researchers usually work on projects, proposed and developed in conjunction with public and private organizations; further, they can be involved in other activities, such as teaching at university. For each scientific theme of interest, there is a research area with one responsible; likewise, for each research project, there is a project team. A researcher can be engaged in more project groups and can be tied to different research themes.

As it is natural within a research organization, collaborative work takes place through a number of communities and work groups. Indeed, in addition to formalized groups (sections, internal committees, research groups, project groups) many informal groups may arise, often spontaneously, around common problems, interests and objectives (e.g., a research topic, a publication, a technology, bureaucratic and administrative issues and so forth) and they may evolve in time. Such groups might be also spread over different locations of ICAR-CNR. In summary, the institute behaves as a complex system, consisting of various kinds of components and actors with a high level of cooperation.

This complex, hierarchical and cooperative structure perfectly fits with *Borè* features. Web portal modeling starts with the graph root, an instance of the *Institute* custom type. This type, that obviously extend *Community*, is equipped with a custom view that will display the portal home page (see Fig. 2.8(a)). The Institute instance is linked to a set of resources, that are automatically grouped according to their category (events, publications, research projects and so forth); people that col-



(a) Institute home page



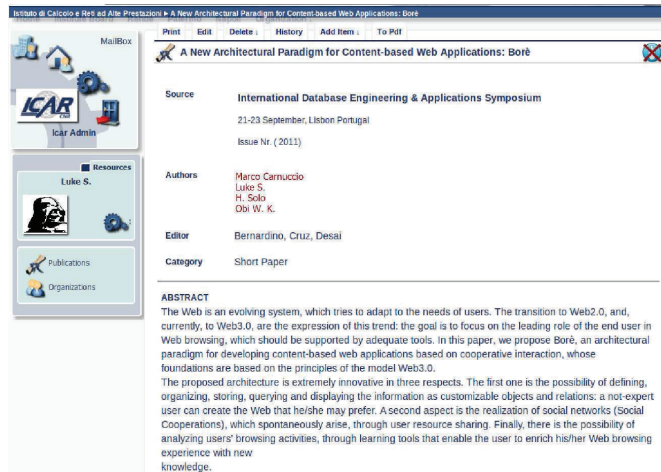
(b) Personal page

Fig. 2.8: ICAR-CNR web portal community pages (www.icar.cnr.it)

laborates with the institute and relevant news/events are shown in other boxes. Each location (Rende, Napoli, Palermo) is an instance of *Community* type; people who work/collaborate are directly linked with their head office. A community instance is also defined for each business role involved (researchers, technicians, professors, etc.). This hierarchical, community-based, structure allows fruitful resource sharing and discussions at any hierarchical level: bottom-up (from specific roles to the whole institute) and top-down (from the institute to all specific roles) sharing approaches are both allowed.



(a) Publications list



(b) Publication details

Fig. 2.9: ICAR-CNR web portal resource pages (www.icar.cnr.it)

For each scientific theme of interest (Database and Knowledge Base Systems, Models and Tools for Parallel Processing etc.) it is defined a community (i.e., a research area) with one *owner* (i.e., the responsible); likewise, for each research project, there is a project team (i.e., *members* of the correspondent community). *Borè* data schema allows each researcher to be involved in more project groups and in different research themes.

A *Publication* is a special, custom type which defines some specific fields (e.g., *year*, *category*, *abstract*) and that can be involved in some specific relations (e.g., *isAuthor*).

Browsing the graph via automatically-generated interfaces (or through the search box on the page top-right) it is possible to reach the node of a PhD Research Fellow¹¹ (i.e., a *member* of the correspondent community), Luke S. (Fig. 2.8(b)). On central page are shown Luke's information while, his related resources (in this case publications and organizations) are listed in left boxes. By clicking on *Publications* link it is possible to browse the list of publication instances linked with Luke through an *isAuthor* relation (i.e., Luke's publications); this list can be ordered and filtered in different ways (see Fig. 2.9(a)). Publications' details (Fig. 2.9(b)) can be investigated by clicking on a single item of the list. Note that boxes on the left are fixed during this browsing operations and they refers to the last community instance visited. If the click stream leads to the visit of another community (e.g., a co-author's community) boxes will be updated with new community's correspondent ones.

For more information about ICAR-CNR institute and an interactive and amazing user experience in a fully functional *Borè* environment, please visit <http://icar.cnr.it>.

¹¹ Fictional for privacy reasons

Caldera: enhancing Borè

3.1 Introduction

In this chapter we propose an extension of the architectural paradigm described in Chapter 2. This extension introduces innovative features with the aim of defining a complete and unified framework named *Caldera*. *Caldera* is a platform for managing and analyzing collaborative process, which integrates an advanced recommender system.

Caldera is innovative under different perspectives, since (i) it allows the definition and the management of semi-structured contents and (ii) to realize Social Networks and Social Cooperations; furthermore (iii) it is integrated with both recommender systems and (iv) with innovative features of process mining through log analysis. As described in Chapter 2, *Borè* already provides an advanced content management system and social networking.

Fig. 3.1 depicts the architecture of the *Caldera* platform, which is composed by three main blocks. The core of the platform is *Borè*; in Fig. 3.1 we only present two macro-blocks with its main features, namely Advanced Content Management and Social Networking. As we aforesaid, these features are the starting point for two extensions. The first one (*Advanced BPM*) allows workflow definition, enactment, analysis and prediction and it will be widely described in Section 3.2. The second one (*Recommendation Engine*) enables knowledge discovery processes in order to infer new knowledge. This knowledge is used to enrich user browsing experience, suggesting new contents of interest. The latter will be analyzed in Section 3.3.

3.2 Advanced business process management

Workflow models (precisely control-flow models) are a popular means for representing process behavior, where all legal ways of executing process activities are specified in terms of precedence constraints and more elaborate routing constructs, such as concurrence, loops, synchronization and choice. In this section, we are going

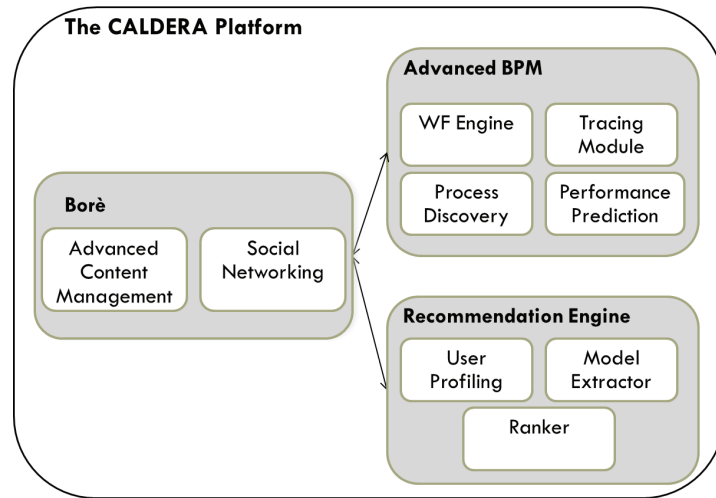


Fig. 3.1: Caldera architecture

to introduce some basic concepts on workflow models and to discuss the *Caldera* BPM block.

A workflow model (a.k.a. workflow schema) specifies all possible flows along the activities of a process, by way of a set of constraints defining “legal” execution in terms of simple relationships of precedence and/or more elaborate constructs such as loops, parallelism, synchronization and choice (just to cite a few). A significant amount of research has been done in the context of specification mechanisms for process modelling (e.g., EPCs [97], Petri Nets [89], and others [37, 84]).

Each time a workflow model is enacted, its activities are executed according to the associated constraints, till some final configuration is reached. Many process-oriented systems store information on process instances in a log repository, keeping track of the events happened during each of them. Basically, a process log can be seen as a set of traces, which, in the most simplistic scenario, correspond to strings over activity identifiers, representing ordered sequences of activities.

Essentially this is the type of historical data that process mining algorithms [92] take in input in order to discover a workflow model, even when the original workflow is unknown. The quality of a workflow model \mathcal{W} can be evaluated relatively to a log \mathcal{L} (the one actually used for inducing the model, or another log of the same process) by way of “conformance” measures (usually ranging over $[0, 1]$), which can be distinguished into two main families: (i) *Fitness measures*, a sort of completeness measures, which roughly tell how much the traces in \mathcal{L} comply with the behavior encoded in \mathcal{W} , by typically counting the violations that are needed to perform to replay all the traces through the model and (ii) *Precision measures*, which try to quantify how much of the flexibility (ascribable to alternative/parallel constructs) of \mathcal{W} is really necessary to reproduce \mathcal{L} .

On the other hand an emerging research stream (see, e.g., [91, 95]) concerns the induction of state-aware models for predicting some relevant performance metrics, defined on process instances. The interest towards such novel mining tools stems from the observation that performance forecasts can be exploited to improve process enactments, through, e.g., task/resource recommendations [83] or risk notification [24]. However, accurate forecasts are not easy to make for fine-grain measures (like, e.g., processing times), especially when the analyzed process shows complex and flexible dynamics, and its execution schemes and performances change over time, depending on the context. In fact, the need to recognize and model the influence of context factors on process behavior is a hot issue in BPM community (see, e.g., [28]), which calls for properly extending traditional approaches to process modeling (and, hopefully, to process mining).

The *Business Process Management* block of the *Caldera* platform exploits *Borè* basic features to construct a powerful, complete and innovative BPM framework, that allows the management of both aspects described above. Fig. 3.2 provides an insight into the whole BPM block that is composed by four different modules:

- *Workflow Engine*. It is the main module of the block. It allows the definition of workflow schema and control-flow models, and it provides the execution environment. The features of this module will be discussed in Section 3.2.1;
- *Tracing Module*. The aim of this module is to generate and export logs as set of traces representing ordered sequences of activities, derived from workflow enactments. Execution traces are enriched by adding additional information that will be useful for process discovery and performance prediction. These logs give information about Process, Activities, Executors and Context Data (as whole system workload or date/time information). Additionally, this module computes, on each trace, the value of specified metrics; the meaning of these metrics will be clarified in Section 3.2.1. Furthermore, the module exports process logs into standard formats as MXML [96] and its extension, the *eXtensible Event Stream* (XES) [98] standard. These standards are recognized from many process mining tools, as PROM [90, 94], a pluggable generic open-source process mining framework. Hence, system administrators can use for log analysis *Caldera* built-in features and/or external process mining tools;
- *Process Discovery*. This module handles modeling of behavioral aspects of business processes. This is an hard and costly task, which usually requires heavy intervention of business experts. In Section 3.2.2 we present a survey of different kinds of techniques that can be exploited in the *Caldera* framework for this purpose;
- *Performance Prediction (AA-TP)*. The aim of this module is to discover predictive models for run-time support. This is, as aforesaid, an emerging topic in Process Mining research, which can effectively help to optimize business process enactments. This topic is widely investigated in Section 3.2.3.

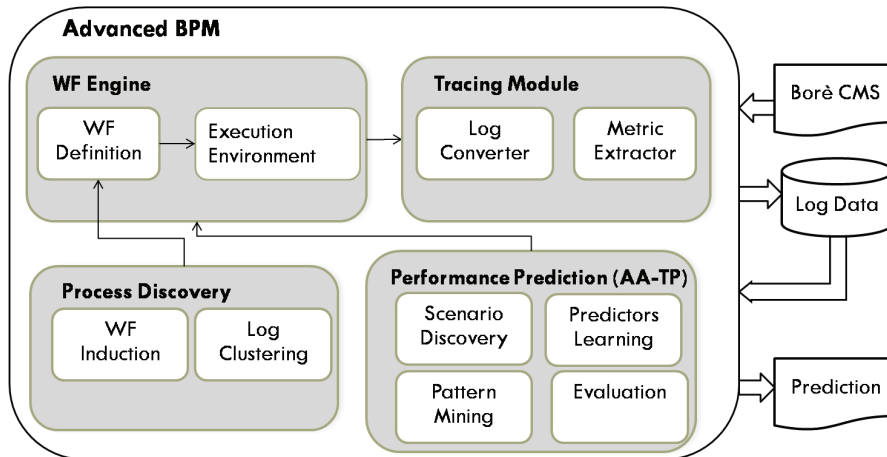


Fig. 3.2: Advanced business process management block

3.2.1 Workflow engine

In this section we will describe the core of the *Advanced BPM* block: the *Workflow Engine*. This module provides tools for workflow schema definition and process enactments. The starting point of the module is the *Borè* content management system. Through some simple extensions of involved taxonomies, it is possible to define adequate resources and relations types. A new resource type taxonomy is reported in Fig. 3.3; the original, basic, taxonomy described in Chapter 2 has been extended with two categories, namely *Process Mining* and *Action*. The *Process Mining* node includes all resource involved in a process schema, as *Activity* (i.e., process activities), *Transition* (i.e., precedence constraints) and *Condition* (i.e., other constraints). The node *Workflow* represents the whole workflow schema (i.e., set of activities, transitions and condition nodes). Last node (*Prediction Metric*) allows the definition of metric measures; a metric is, basically, a measure composed by (i) a name (e.g., remaining time, total cost etc.) and (ii) a procedure to compute it. Each workflow schema can be equipped with one or more metrics, that will be automatically calculated and extracted by the *Tracing Module*. These metrics can be used for process discovery and performance prediction, as discussed next. The *Action* node allows users to define a set of custom action that will be performed after the occurrence of specific events (e.g., end of an activity); as described in Chapter 2, some actions are provided by default (e.g., email notifiers).

In a similar way, Fig. 3.4 depicts the extension of the original relation type taxonomy. This extension provides ad-hoc relations to manage control flow, precedence and other kinds of constraints. There are three main relation involved: *fromTransition*, *toTransition* and *toCheck*. The *fromTransition* relation is useful to connect an activity and the transition that will be activated when the former ends. Symmetrically we need the *toTransition* relation to connect a transition with an activity; this

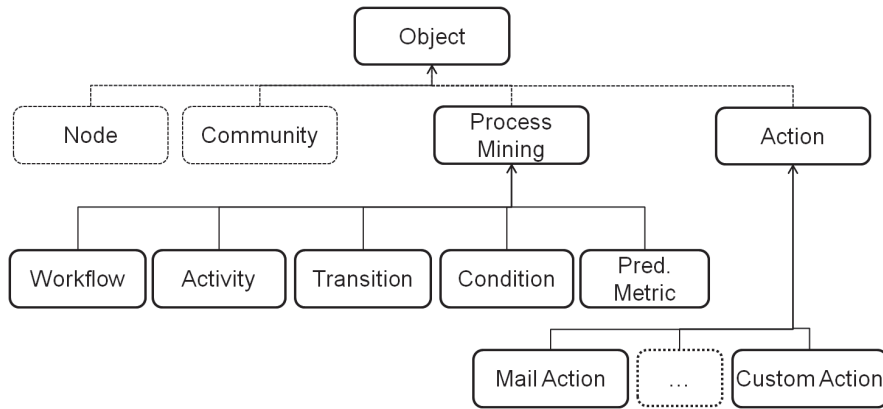


Fig. 3.3: Workflow resource type taxonomy

activity will start when linked transition is activated. Note that, as they are defined in Chapter 2, relations can connect asymmetrically (i.e., with a direction) only two nodes. Last relation, named *toCheck*, is used to connect a transition with a specific condition.

A transition is “active” iff (i) the upstream activity (i.e. connected with the *to-Transition* relation) is finished and (ii) all associated conditions are true (or this set is empty). Dually an activity can start iff all the incoming transitions (i.e., connected with the *fromTransition* relation) are active (or the set of incoming transition is empty).

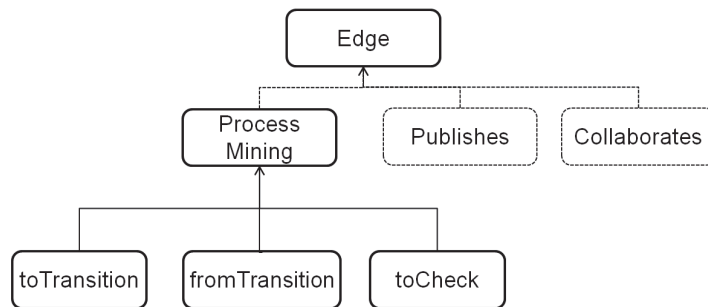


Fig. 3.4: Workflow relation type taxonomy

Fig. 3.5(a) depicts a basic workflow schema while Fig. 3.5(b) shows its conversion into *Caldera* formalism. The workflow consists of four activities and it starts with the one with no incoming transition, *Act1*. At the end of this activity, there are two concurrent ones, namely *Act2* and *Act3*. The former activity can start only if/when the associated condition is true, while the latter starts automatically. Final

activity (no outgoing transitions) Act^F can start when $Act2$ and $Act3$ are *both* finished. The translation of this schema in our formalism is quite simple: each activity, transition and condition is translated into a (correspondent) resource. Then, adequate relations must be inserted to connect these resources. In this example, only one prediction metric (i.e., remaining time) is associated to the workflow schema. For more details about the conversion see Fig. 3.5(b).

3.2.2 Process discovery

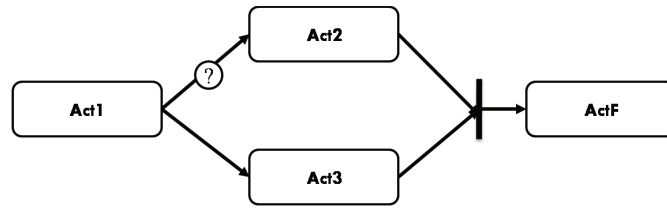
Modelling behavioral aspects of business processes is an hard and costly task, which usually requires heavy intervention of business experts. This explains the increasing attention given to process mining techniques, which automatically extract behavioral process models from log data. This section presents and offers a survey on different kinds of basic techniques that can be exploited to this purpose; these can be classified into two categories: (i) workflow induction and (ii) log clustering. Note that this section is a pure state-of-the art presentation. Implementation details in the *Caldera* framework of these well-known techniques are not in the purpose of this thesis.

Workflow induction

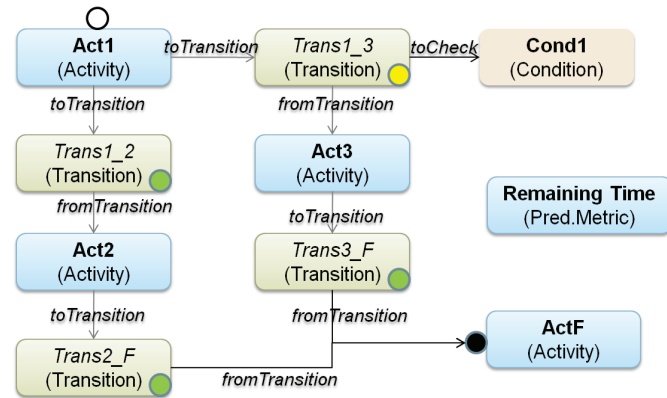
This subsection describes a series of techniques specifically designed for workflow schemas induction. In general, these techniques attempt to automatically extract, for a given set of execution instances, a workflow model that describes, in a concise and enough accurate way, the main flow actually performed. Clearly, such model represents a precious tool to analyze the real behavior of a given process (for comparison with existing models) or to improve future executions.

The general problem of discovering a structured workflow model was analyzed in [93], to capture a class of Petri nets that can be rediscovered via induction methods. The algorithm proposed, named α , can rediscover such a model, under the hypothesis that the input log is complete (i.e., all pairs of tasks linked directly in the workflow appear consecutively in at least one log trace). Two extended versions of this algorithm [30, 104] were proposed subsequently to discover two specific kinds of control-flow constructs: short loops (loops involving one or two activities only) and non-free-choice constructs (where the choice of which outgoing edges of a XOR-split node x is to be executed does not depend on x only), respectively. Simple metrics concerning task dependency and task frequency are exploited in a heuristics approach [103] capable of discovering a graph-based model, called “dependency/frequency graph”. Notably, this approach can cope with noisy logs, based on user-given frequency thresholds. The discovery of block-structured workflows (specified with the ADONIS [60] language) possibly containing duplicate tasks was addressed in [50, 51], where a two-step solution is presented: first a stochastic activity graph is induced from the log, and then it is turned into a block-structured workflow by suitable transformation rules. The use of term rewriting systems was also proposed to discover a hierarchically structured workflow model, in the form of an expression tree, where the leafs represent tasks while any other node is associated with a control flow operator [82]. In [23] has been devised an approach which

adopts the paradigm of planning and scheduling by resource management to tackle the combinatorial complexity of the problem.



(a) Workflow basic schema



(b) Workflow engine representation

Fig. 3.5: A simple workflow schema and its representation in the workflow engine module

An alternative solution [31] to the workflow discovery problem adopts a global search approach, based on genetic algorithms. This allows for dealing with complex routing constructs (including non-free-choice and hidden tasks, i.e., routing activities that do not appear in the traces) and with noisy data, but implies highest computational costs.

The effectiveness of a workflow discovery algorithm can be evaluated according to various quality dimensions, which include the fitness and precision measures mentioned in Section 3.2. Maximal fitness can be easily achieved by all the algorithms above. However, this does not imply that the resulting model really captures the possible behavior of the unknown process, if the log does not satisfy the completeness notion underlying the induction algorithm. In particular, most algorithms based on local search [30, 73, 104], assume adjacent tasks appear consecutively in some log traces. This discourse gets more varied when the process follows complex control-flow constructs (non-free-choices, duplicate tasks, hidden tasks) and logs are noisy. The precision of process mining algorithms may rapidly fall when the analyzed process exhibits different execution scenarios, possibly combined with global

behavioral constraints. In such a case, good results are achieved by approaches based on genetics algorithms [31] or on clustering [40]. On the other hand, the first kind of approach might be computationally unviable for large logs, while an excessive partitioning of log traces may lead to overfitting. In fact, more generally, the size of the log can impact severely on the real value of a discovered process model, especially when the analyzed process exhibits complex dynamics and a high level of concurrency. Indeed, in such a case, small samples of log traces hardly capture the different sequencing of activities that are admitted for the process, so that the model eventually discovered is likely to provide an under-generalized (“overfitted”) representation of the process behavior. For example, it may happen that a precedence is incorrectly discovered between activities belonging to mutually parallel branches of the process, only because, in the given (incomplete) log, these activities always appear in the same order. A possible way to somewhat prevent the generation of overfitted models, in the case of clustering-based methods, is to simply set an upper bound to the number of clusters. Anyway, using abstraction mechanisms as a pre-processing or post-processing tool can help alleviate this problem. For example the method in [44] provides the analyst with a simplified dependency graph, where only significant enough activities and edges are depicted, while omitting (or aggregating) minor structural elements of the process structure.

Log clustering

Clustering techniques can help recognizing different behavioral classes of process instances automatically, by exploiting the information captured in log data. In this subsection we analyse some methods for trace clustering recently proposed in literature.

A prevalent approach to clustering traces consists in transforming traces into vectors where each dimension corresponds to an activity [29, 40, 87]. Different methods were proposed to project log traces into such a feature space, most of which focus on the frequency of activities in the log traces. Clearly such a bag-of-activities representation, suffers, as a major drawback, from the loss of temporal information, in that it does not account for the ordering of activities. One way for alleviating this problem is to regard each trace as a sequence of activities and to extract a number of *k-grams* (i.e., subsequences of length k) from it, as features for the clustering [87].

In particular, in [87], the vector space model is used with multiple feature types, corresponding to different trace profiles, i.e., sets of related items describing traces from a specific perspective (activities, transitions, data, performance, etc). Each item is associated with a measure assigning a numeric value to any trace. Therefore, by transforming each log trace into a vector containing all these measures, any distance-based clustering method can be exploited to partition the log. In particular, three distinct distance measures are considered to calculate the similarity between cases: *Euclidean* distance, *Hamming* distance and *Jaccard* distance. Using these similarity measures, different clustering schemes are exploited applied to partition log traces (e.g., K-means, Agglomerative Hierarchical Clustering, etc.).

This vector space model was combined with new context-aware features [58], by expanding the core idea of considering activity subsequences conserved across mul-

tuple traces. Unlike the *k-grams*, subsequences of variable length are detected which frequently occur in the log, and are assumed to correspond to some hidden functionalities of the process. Using these conserved subsequences as features, the clustering is expected to put together traces that are similar from a functional viewpoint.

A hierarchical clustering approach exploits a special kind of sequential features, named *discriminant rules* [40], devised for capturing behavioral patterns that are not properly modeled by a given workflow model. Precisely, a *discriminant rule* has the form $[a_1 \dots a_h] \not\rightarrow a$ s.t.: (i) $[a_1 \dots a_h]$ and $[a_h a]$ are both “highly” frequent (i.e., the frequency is above a given threshold σ), and (ii) $[a_1 \dots a_h a]$ is “lowly” frequent (its frequency is below another threshold γ). Such rules can be straightforwardly derived from frequent sequential patterns, discovered efficiently via a level-wise search strategy [2, 3, 46].

A different approach to trace clustering dismisses the vector space model in favour of syntactic techniques which operates on the whole sequence “as-is” by way of string distance metrics. For instance, a context-aware approach based on the generic edit distance [78] was proposed in [58]. The edit distance between two sequences is defined as the cost of the optimal combination of edit operation (insertion, deletion or substitution) that allows one sequence to be transformed into another. The cost of edit operations can be adapted to the peculiarities (primarily, concurrence nature) of workflow processes by devising ad-hoc algorithms [58] for automatically deriving an optimal setting of such costs.

The main drawback of string-oriented techniques [58] is the typically higher computational cost, which may make them unpractical when massive logs are to be analyzed. Notice that this problem cannot be circumvented, in general, by way of sampling techniques, as a huge number of distinct log traces can be actually necessary to rediscover a workflow with many parallel branches – which may lead to many distinct log traces that only differ from each other in the ordering of the parallel (and hence mutually independent) tasks. And yet, heuristics-based correction mechanisms for taking account of the concurrent nature of workflow processes, might be ineffective against highly concurrent processes. In principle, higher scalability is achieved with feature-based approaches [29, 40, 59, 87], owing to the possibility to exploit consolidated efficient methods for the clustering vectorial data and to the exploitation of efficient algorithms for deriving the features from the given log. On the other hand, the quality of results depend on the capability of the considered structural patterns to capture and discriminate the main execution variants of the process. Hence, a trade-off between the expressiveness of the patterns used as features and the cost of extracting them must be suitably selected, according to the specific application context.

As a final remark, we believe that the clustering of log traces could well take advantage of the brilliant results achieved in the field of co-clustering, one of the hottest topics in Data Mining community in recent years, where multiple data types are to be partitioned simultaneously based on their mutual correlations. Indeed, co-clustering methods have shown brilliant results even when the goal is to cluster one data type, associated with a sparse and high-dimensional space of attributes (like, e.g., text documents and associated terms). A pioneering effort along such a direction was recently

done in [38] (in an outlier detection setting), where the mining of structural patterns is combined with a co-clustering scheme focusing on the associations between such patterns and the given log traces. In our opinion, such an approach can achieve good quality results when used for clustering the log of a process featuring a large number of structural patterns, without incurring in the notorious “curse of dimensionality” problem.

3.2.3 Performance prediction

Discovering predictive models for run-time support is an emerging topic in Process Mining research [91,95], which can effectively help optimize business process enactments. In general, making accurate estimates is not easy especially when considering fine-grain performance measures (e.g., remaining processing time) on a complex and flexible business process, where performance patterns change over time, depending on both case properties and context factors (e.g., seasonality, workload).

In general, performance estimations on process execution have a wide range of applications:

- *Additional information.* Estimations can be simply shown to final user as run-time information. User is always aware about process enactment and estimated remaining time. Furthermore, this information is very useful for system administrators: an aggregate view of predictions about active processes in the whole system can be used in order to analyze long-term workload and to optimize resources allocation;
- *Decision support.* Defined (and predicted) metrics can be seen as additional attributes of the process. These forecasted attributes may be used in workflow basic schema to express some data-driven control flow rules, in which “virtual” attributes can be useful to identify, in split points, the most suitable paths to follow;
- *Recommendation.* In some real applications, where workflow models are flexible and not well defined, a hidden workflow model can be induced through predictions [83]. An advanced BPM can suggest to final user the best path, through ranking all possible next activities. System, in fact, can simulate possible evolutions of the process by adding (one by one) all allowed activities; then, for each evolution, metrics are forecasted and the one with the better predicted value can be proposed as the only next activity in the flexible workflow schema;
- *Risk analysis.* Knowing in advance such metrics, as the remaining processing time, is a great value for system administrators in order to prevent violations of SLA (service level agreement). Certain typical SLAs, in fact, establish that process enactments must not last more than a maximum fixed time; otherwise pecuniary penalties may be charged to administrators [24].

In [8,9] we introduced a novel approach to the discovery of predictive process models. To this purpose, we combine a series of data mining techniques (ranging from pattern mining, to non-parametric regression and to predictive clustering) with

ad-hoc data transformation and abstraction mechanisms. As a result, a modular representation of the process is obtained, where different performance-relevant variants of it are provided with separate regression models, and discriminated on the basis of context information. Notably, this approach is capable to look at the given log traces at a proper level of abstraction, in a pretty automatic and transparent fashion, which reduces the need for heavy intervention by the analyst (which is, indeed, a major drawback of previous solutions in the literature). Our innovative approach has been validated on a real application scenario, with satisfactory results, in terms of both prediction accuracy and robustness. Approach presented in [8] has been awarded with the *Best Paper Award* during the 15th International Conference on Enterprise Information Systems (ICEIS13).

The use of prediction techniques was first considered in [69] with the aim of capturing the impact of workload-oriented context features on the performances of the resources involved in the execution of a single work item (in terms of expected “service times”). An emerging research stream in the area of Process Mining concerns the induction of models for forecasting performances metrics of a whole process instance [36, 91, 95], rather than of a single resource.

The task of predicting the values of a numeric (or continuous) target variable is known as regression in statistics. Basically, regression abstracts from a set of measures by searching the function that best fits some available values. In literature is usual to categorize regression in: (i) *parametric regression* [48], where the function is assumed to have a certain form (e.g., linear, exponential, quadratic), and (ii) *non-parametric regression*, where no a-priori assumption are needed about the kind of regression function fitting the values [47]. Among the parametric regressors, the linear one is undoubtedly the most popular modeling approach because of its simplicity. Actually, a linear predictor [35] models the relationship between a target variable and one (or more) descriptive variables by means of a linear function whose parameters are estimated from training data. Clearly, linearity is the main drawback of these models since they fail to fit well over data exhibiting a non-linear dependency. Non-parametric regressors can be further classified in *eager* (or model-based) and *lazy* (or instance-based) regressors. Eager are those regression systems that induce a model during the training. This model is next used to interpret the underlying data. Lazy approaches, instead, do not construct models and postpone processing to the prediction phase. Tree-based regressors are an example of eager approaches, where numeric predictions are enabled by storing in each leaf either a constant (typically, the average) or a regression model providing a good approximation to the distribution of the target variable across the instances fallen into the leaf. The former case corresponds to the *regression trees* (e.g., *CART* [18] and *RepTree* [105]) whereas the latter corresponds to *model trees* (e.g., *M5* [75], where the underlying model takes, indeed, the form of a linear function). *CART* builds the tree by recursively splitting the training set, while examining, at each iteration step, all descriptive variables and determining a split that help minimize the variance in the target variable. The process is recursively repeated on the portions of data resulting from the first split until homogeneous terminal nodes are achieved.

A similar strategy (using either the information gain or the variance as the variability measure to minimize) is undertaken by *RepTree*, where the grown tree is eventually pruned via the reduced-error pruning method. Optimized for speed, it deals with missing values by splitting instances into pieces, as C4.5 [74] does. The model tree *M5* shares with CART the same splitting heuristic (i.e., variance reduction). It first learns a standard regression tree, and then turns it into a model tree during the pruning phase. In practice, each internal node is replaced with a leaf containing a linear model, provided that the model performs at least as good as the subtree rooted in that node.

As pertaining the lazy regressors, a simple and yet very popular method is the k -nearest neighbor (k -nn) one. In k -nn, the target variable is predicted by weighting the k nearest observations in the training set. The closeness among instances is defined in terms of similarity w.r.t. their descriptive variables. The kind of distance measure, the weighting scheme as well as the number of neighbors are all selectable options for a k -nn regression schema. The main drawback of k -nns (and of lazy regressors, in general) is that they have to store and search over a large set of examples, resulting in high memory requirements and prediction times; in principle, this limitation could be partly alleviated by resorting to suitable sampling methods. As an advantage, instead, they are easier to implement and faster in the learning phase (since, in fact, they do not need to generalize data by way of any sort of model).

Besides classical regression models, numeric prediction can be also done by means of some clustering schemes. The core idea of Predictive Clustering approaches [14] is that, once discovered an appropriate clustering model, a prediction for a new instance can be based only on the cluster where it is deemed to belong, according to some suitable assignment function. The underlying belief is that the higher similarity between instances of the same cluster will help derive a more accurate predictor w.r.t. the one induced from the whole dataset. Specifically, two kinds of features are considered for any element z in the given instance space $Z = X \times Y$: *descriptive* features and *target* features (to be predicted), denoted by $descr(z) \in X$ and $targ(z) \in Y$, respectively. Then, a *Predictive Clustering Model (PCM)*, for a given training set $L \subseteq Z$, is a function $q : X \rightarrow Y$ of the form $q(x) = p(c(x), x)$, where $c : X \rightarrow \mathbb{N}$ is a partitioning function and $p : \mathbb{N} \times X \rightarrow Y$ is a (possibly multi-target) prediction function. Clearly, whenever there are more than one target features, q encodes a multi-regression model. Several PCM learning methods have been proposed in the literature, which can work with general relational data [14], or with propositional data only (e.g., system CLUS [33]). An important class of such models are Predictive clustering trees (PCTs) [14, 15], where the cluster assignment function is encoded by a decision tree. The tree-based structure is induced in a top-down fashion by recursively partitioning the training set. A variety of PCT learning methods exists in the literature, which differ in the type/number of target features (e.g., decision trees, regression trees, multi-target regression models, clustering trees), or in the underlying representation of data instances either relational (e.g., system TILDE [14]) or propositional (e.g., system CLUS [33]). Since the structure of process logs (performance-annotated sequences of events) makes a trivial application of PCT learning methods ineffective and/or computationally expensive, an ad-hoc propositional encoding of

the log trace was devised in [36], where each trace is mapped into a tuple featuring all its context properties and a heuristically selected set of performance measurements.

All of these approaches share the idea of abstracting a trace into a concise description of the activities executed during its unfolding. Typically, each trace is converted into a collection (e.g., set, multiset, list or vector) of some of the properties associated with its events¹. For instance, one could simply turn a trace into the set of tasks executed in it. In general, the definition of a suitable trace abstraction function, especially in the case of complex business processes, capable to focus on the core properties of the events (happened in a process instance) that impact the more on its performance outcomes, is a critical issue. In fact, as discussed in [91], choosing the right abstraction level is a delicate task, where an optimal balance has to be reached between the risks of overfitting (i.e., having an overly detailed model, nearly replicating the training set, which will hardly provide accurate forecasts over unseen cases) and of underfitting (i.e., the model is too abstract and imprecise, both on the training cases and on new ones).

Based on such abstract representations of log traces, a special class of annotated state machines (named A-FSM) is built in [91], where each node corresponds to exactly one abstract trace representation and stores an estimate for the target measure. By equipping each node s with some statistics (e.g., the average) derived from the performance values of all the trace prefixes sharing the same abstract view as s , it is possible to eventually make continuous predictions for any new instance of the process; at any moment, the forecast for a new process instance is estimated via the statistics stored in the node currently reached by it in the model. In such a way, the model will feature as many states as the distinct abstract abstractions of the log traces. A non-parametric regression model is used in [95], where the performance for a new (possibly partial) trace is estimated based on its similarity to historical ones. To this purpose, some kernel-based functions are used, defined over a vectorial representation of the traces, encoding occurrences and, possibly, activity durations and case-data variables; like in [76], these functions are set in a data-driven fashion, by automatically setting their associated bandwidth factors based on a cross-validation procedure. These measures will be eventually exploited to make a forecast for a new trace τ , according to an instance-based regression scheme, where the performance value of τ is estimated as a weighted sum of the performance values of all the traces in the training log (the closer to τ an example trace is, the higher its weight in the summation). The approach in [36] extends the basic learning technique in [91] in order to efficiently and effectively take account for the dependence of process performances on context factors. The approach introduces an ad-hoc predictive clustering method, where different context-related execution trace clusters (or *process variants*) are discovered for the process, and provided with separate A-FSM models. In this way, a forecast for a new process instance is made by using the A-FSM model of the cluster it is estimated to belong to.

¹ Basically, each log trace stores, for each event occurred during a process case, a series of properties (e.g., the executed task and the executor).

A major drawback of FSM-based approaches [36,91] lies in the fact that the size of the discovered (annotated) transition model may explode in the case of complex logs, featuring a high number of high-level event abstractions (e.g., process tasks), and of paths across them (state nodes may be combinatorial in the number of distinct event abstractions). In this way, the model may become overly complex, besides risking overfitting the training log and producing inaccurate predictions on new (unseen) process instances. This problem can be alleviated by allowing the increase of the abstraction level over traces, e.g., by fixing a horizon threshold h on past history (i.e., only the latest h events appearing in each trace are kept in the abstracted view). However, in general, it may well be very difficult to find a good level of abstraction guaranteeing a satisfactory trade-off between generalization and accuracy. On the other hand, FSM models cannot exploit effectively non-structural properties of the process instances, which might well be relevant for the prediction of performance outcomes. In fact, the idea of including such data (in addition to executed tasks) in the construction of trace abstractions will clearly emphasize the combinatorial explosion issue discussed above. The problem of properly setting the abstraction level does not affect the approach in [95], where different vectorial representations of the traces are used to define different kernel-based (and self-tuned) distance measures. However, in general, such an instance-based scheme cannot ensure a fast enough computation of predictions, actually needed for providing an effective run-time support to many real BPM platforms, especially in the case of a complex and flexible process, where a large amount of historical traces should be maintained to capture adequately its wide range of behaviors.

In [8,9] we overcome the above limitations, by devising a novel approach, capable of both taking full advantage of “non structural” context data (usually stored for each process instance, and often strongly correlated with its performances) and of finding a good level of abstraction over on the history of process instances, in a pretty automated and transparent fashion. In these works we showed that handy (and yet accurate enough) prediction models can be learnt via various existing model-based regression algorithms (either parametric, such as, e.g., [48,75], or non-parametric, such as, e.g., [47,105]), rather than resorting to an explicit representation of process states (like in [36,91]) or to an instance-based approach, like in [95]. This clearly requires that an adequate propositional representation of the given traces is preliminary build, capturing both structural (i.e., task-related) and “non-structural” aspects. Our proposition is to convert each process trace into a set or a multi-set of process tasks, and let the regression method decide automatically which and how the basic structural elements in such an abstracted view of the trace are to be used to make a forecast.

Technically, we extend and integrate the method for inducing predictive performance models presented in [91] and the logics-oriented approach to predictive clustering [15], where the discovered model takes the form of a decision-tree. The discovery of such scenarios (i.e., clusters) is carried out by partitioning the log traces based on their associated context features, which may include both internal properties of a case (e.g., the amount of goods requested in an order management process)

and external factors that characterize the situation where it takes place (e.g., workload, resource availability and seasonality indicators).

Moreover, we still leverage the idea of [36], of combining performance prediction with a predictive clustering technique [14], in order to distinguish heterogeneous context-dependent execution scenarios (“variants”) for the analyzed process, and eventually provide each of them with a specialized regressor. In general, such an approach brings substantial gain in terms of readability and accuracy, besides explicitly showing the dependence of discovered clusters on context features, and speeding up (and possibly parallelize) the computation of regression models. In fact, these models are typically more compact, more precise and easier to read/evaluate/validate than a single regression model extracted out of the whole log. Our core idea is that also very simple regression methods can furnish robust and accurate predictions, if combined with a properly devised clustering procedure. Target features used in the clustering (where context features conversely act as descriptive attributes) are derived from frequent structural patterns (still defined as sets or bags of tasks), instead of directly using the abstract representations extracted by the log, as done in [36]. These patterns will be discovered efficiently via an ad-hoc *a-priori* like [3] method, where the analyst is allowed to specify a minimum support threshold, and possibly an additional “gap” constraint, both enforced in the generation of the patterns. Notably, such an approach frees the analyst from the burden of explicitly setting the abstraction level (i.e., the size of patterns, in our case), which is determined instead in a data-driven way. Results from an experimentation on a real-life application scenario (pertaining the handling of containers in a maritime terminal) are encouraging, in that they show that the method exhibits good accuracy and robustness, and it requires little effort in the setting of its parameter; in particular, it suffices not to use extreme values for pattern support threshold, to ensure very low prediction errors, no matter of the other parameters. This clearly matches our general goal of providing the analysts with a easy-to-use (data-adaptive) prediction technique.

A detailed explanation about proposed approach, the prototype system AA-TP (Adaptive-Abstraction Time Prediction), experiments setting and advantages discussion are not in the aim of this thesis and can be found in [8, 9].

3.2.4 A case study: University of Calabria internships platform

In this section we are going to present a real-life case study that exploits *Caldera* BPM features: the University of Calabria internships platform.

An university internship consists of a period of training and guidance, promoted for students enrolled in various courses of study at the University of Calabria. The apprenticeship training and guidance is aimed at creating fruitful moments of alternation between study and work in the educational processes and to facilitate career choices through direct knowledge of business world. Training is intended for students who have already completed their educational activity. The University Internship has to be held at public or private institutions, companies, associations, foundations, consortia, professional offices, businesses and Industries (hereafter referred to as Companies/Institutions) that have entered into a special convention with the University.

University of Calabria provides this service since 1998 for all students enrolled in degree courses and it plays a central coordination role of the training activities provided for students.

University of Calabria campus is equipped with several departments that, in Internships scenario, perform the following functions:

- Students front-office;
- Managing relationships with Companies/Institutions;
- Signing Agreements and Training Projects.

The main internships office, instead, holds the following roles:

- Transmits copy of the Agreements and of Training Projects to the Regional and Provincial authorities;
- Stores Agreements and Training Projects;
- Processes and transmits information and data relating internships to the Regional and Provincial authorities.

During past years the need for a complete informative system has increased exponentially. This system should manage and monitor all involved activities (signing of the Agreement, structuring of the Training Project and so forth) to ensure an efficient and effective service and it should offer to students good opportunities to enter and to keep in touch with business world. This integrated system should be able to realize: (i) a single Agreement model; (ii) a unique pattern for Training Project; (iii) an integrated data source of Companies/Institutions joint with the University; (iv) an integrated data source of training projects and conventions; (v) a platform in which each involved actor (Students, Companies, Departments and Internship Office) can play his role.

Fig. 3.6 depicts workflow schemas for the University Internship system. Whole system has to manage two different scenarios (i) Agreement and (ii) Internship that are sketched in Fig. 3.6(a) and 3.6(b), respectively. Activity names are highlighted in bold, while in brackets we point out actor(s) involved in each activity.

First schema (Fig. 3.6(a)) describes the Agreement process enactment. Process starts with a Company/Institution that requests an Agreement for a specific Department. After several steps (including two-way signature and data transmission to external offices), agreement can be rejected (e.g., for lack of requirements) or, finally, registered and stored. All steps are sequentially related (i.e., at most one activity at a time is running and each activity can start only when previous ones are completed).

Internship process enactment is depicted in Fig. 3.6(b). Process starts with a Student that requests an internship, selecting a set of relevant companies/institutions, among available ones. If the student's home department approves the Internship, workflow proceeds with signatures, data transmissions and so forth. This schema is a little more complicated than the other one; during the internship period, in fact, it is possible to manage some requests: extend (i.e., increase duration), suspend (i.e., temporary stop) or interrupt (i.e., stop and cancel). Each of these requests must be approved by the reference department and, if accepted, it induces new data transmissions to external offices.

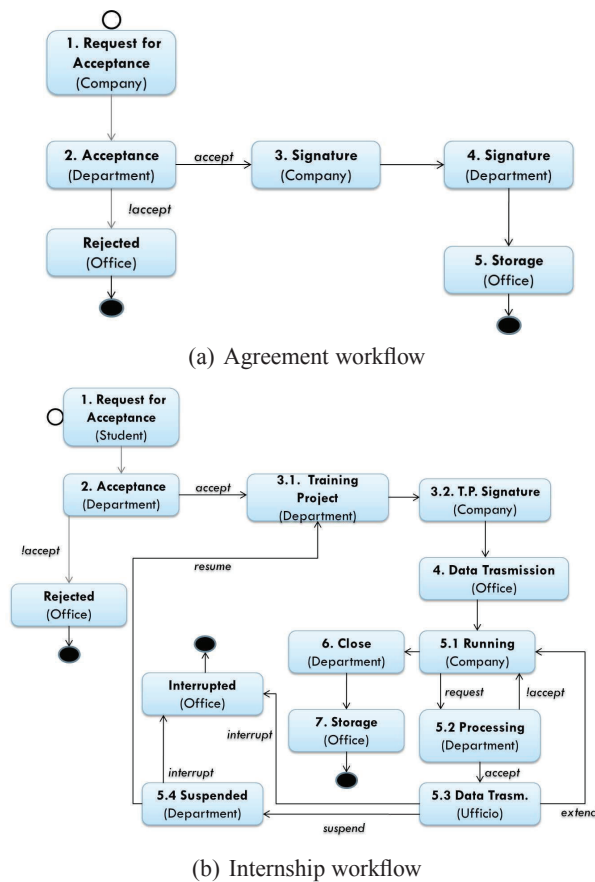


Fig. 3.6: University internships workflow schemas

Caldera represents the most effective framework to use for the depicted system. In particular, Advanced Content Management System (inherited from *Borè*) and Business Process Management module give a simple and ready-to-use solution to this purpose. The *University of Calabria Internship Platform* is currently in a private beta version² available for Main Internship Office and for Departments of the University of Calabria. The workflow scenarios have been deployed in the platform simply translating schemas depicted in Fig. 3.6 into framework formalism — in Section 3.2.1 we shown how to translate classic workflow schemas into *Caldera*-compliant instances (see Fig. 3.5)) — while involved and specific contents (i.e., Agreement, Training Project, Data to be transmitted) have been easily defined in the flexible *Borè* environment. Estimated benefits of the platform are significant: an integrated system in which actors can play their specific role and can monitor,

² <http://160.97.4.193>

real-time, the enactment of running instances. Furthermore, *Performance Prediction* functionalities make the platform extremely innovative: for each process instance (Agreement or Internship) remaining processing time (or other performance parameter to be defined) can be estimated; this opens a completely new scenario for organizational and bureaucratic authorities.

In Fig. 3.7 are reported some screenshots of the *University of Calabria Internship Platform* as it appears in his private beta version. Enactments of Agreement and Internship workflow are reported in Fig. 3.7(a) and Fig. 3.7(b), respectively. Fig. 3.7(c) depicts an example of reporting functions available for main internship office; in this case number of distinct agreements and internships over a specific year, group by department.

3.3 Recommendation engine

With the increasing volume of information, products, services (or, more generally, items) available on the Web, the role of *Recommender Systems (RS)* [77] and the importance of highly-accurate recommendation techniques have become a major concern both in e-commerce and academic research. In particular, the goal of a RS is to provide users with not trivial recommendations, that are useful to directly experience potentially interesting items. Moreover, their exploitation in e-commerce can also provide more interactions between the users and the system, that can be profitably exploited for delivering more accurate recommendations. RSs are widely employed in different contexts, from music (Last.fm³) to books (Amazon⁴), movies (Netflix⁵) and news (Google News⁶ [27]), and they are quickly changing and reinventing the world of e-commerces [81]. Recommendation can be considered as a “push” system which provides users with a personalized exploration of a wide catalog of possible choices. While in a search based system the user is explicitly required to type a query (what he is looking for), here the query is implicit and corresponds to all the past interactions of the user with the system (items/web pages previously purchased/viewed). Recommendations, as introduced before, help users in exploring and finding interesting items which the user may not found on his own. By collecting and analyzing past users’ preferences in the form of explicit ratings or like/dislike products, the RSs provide the user with smart and personalized suggestions, typically in the form of “Customers who bought this item also bought” or “Customers who bought items in your recent history also bought”. The goal of the provider of the service is not just to transform a regular user in a buyer, but also make his browsing experience more comfortable, building a strong loyalty bond .

The strategical importance of the development of always more accurate recommendation techniques has motivated both academic and industrial research for over

³ <http://last.fm>

⁴ <http://amazon.com>

⁵ <http://netflix.com>

⁶ <http://news.google.com>

Richiesta di accreditamento

Ente richiedente: [OpenKnowTech srl](#)
 Dipartimento richiesto: **Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica (DIMES)**
 Data Richiesta: **18 Febbraio 2013**
 Data Stipula Convenzione: **20 Maggio 2013**
 Numero Repertorio: **23/2013**
 Data Reperitorio: **01 Giugno 2013**
 Numero Massimo di Tirocinanti: **4**
 Attività Tirocinanti: **Sviluppo di applicazioni mobili**

- MANIFESTAZIONE DI INTERESSE** DATA AVVIO: 18/02/2013
Visualizza Manifestazione di Interesse
- ACCETTAZIONE DOMANDA** DATA AVVIO: 01/03/2013
La richiesta è stata accettata.
- DOCUMENTO CONVENZIONE** DATA AVVIO: 20/03/2013
Convenzione Firmata Ente
- DOCUMENTO CONVENZIONE** DATA AVVIO: 08/04/2013
Convenzione Firmata Dipartimento
- PROTOCOLLO** DATA AVVIO: 20/05/2013

(a) Agreement process enactment

Tirocinio

Studente: **87056 - Alessandria Serena**
 Ente Ospitante: [OpenKnowTech srl](#)
 Dipartimento: **Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica (DIMES)**
 Numero Pratica: **239**
 Data Richiesta: **04 Luglio 2013**
 Numero Mesi: **3**
 CFU: **5**
 Data Inizio: **01 Settembre 2013**
 Data Fine: **30 Novembre 2013**

- RICHIESTA** DATA AVVIO: 04/07/2013
La richiesta è stata accettata.
- COMPILAZIONE PROGETTO FORMATIVO** DATA AVVIO: 14/07/2013
Visualizza Progetto Formativo
- COMUNICAZIONE ENTI** DATA AVVIO: 24/07/2013
Comunicazioni Correttamente Inoltrate
- TIROCINIO IN CORSO** DATA AVVIO: 01/09/2013
Attestazione fine tirocinio
- CHIUSURA TIROCINIO** DATA AVVIO: 30/11/2013
Tirocinio Chiuso dal Dipartimento.

(b) Internship process enactment

Numero convenzioni per Anno Accademico 2012/2013

Dipartimento	Numero convenzioni
Dipartimento di Biologia, Ecologia e Scienze della terra (DiBEST)	12
Dipartimento di Chimica e Tecnologie Chimiche	12
Dipartimento di Farmacia e Scienze della Salute e della Nutrizione	13
Dipartimento di Fisica	12
Dipartimento di Ingegneria Civile	74
Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica (DIMES)	74
Dipartimento di Ingegneria per l'Ambiente e il Territorio e Ingegneria Chimica	74
Dipartimento di Lingue e Scienze dell'Educazione	87
Dipartimento di Matematica e Informatica	12
Dipartimento di Scienze Aziendali e Giuridiche (DISCAG)	111
Dipartimento di Scienze Economiche, Statistiche e Finanziarie (Di.S.E.S.F.)	111
Dipartimento di Scienze Politiche e Sociali	10
Dipartimento di Studi Umanistici	87

Numero Tirocini per Anno Accademico 2012/2013

Dipartimento	Numero tirocini
Dipartimento di Biologia, Ecologia e Scienze della terra (DiBEST)	42

(c) Reporting functions

Fig. 3.7: University of Calabria internships platform

15 years; this is witnessed by huge investments in the development of personalized and high accuracy recommendation approaches. On October 2006, Netflix, leader in the movie-rental American market, released a dataset containing more of 100 million ratings and promoted a competition, the *Netflix Prize*⁷ [7], whose goal was to produce a 10% improvement on the prediction quality achieved by its own recommender system, *Cinematch*. The competition lasted three years and was attended by several research groups from all over the world, improving and inspiring a fruitful research.

In the *Borè* platform the social cooperation environment leads to a really interesting aspect: a lot of information is shared among users. This information can be used, as described above, to activate knowledge discovery processes in order to infer new knowledge and to enrich users' next browsing experiences. This is the aim of the *Recommendation Engine* block in the *Caldera* platform. Exploiting *Borè* basic features, it provides a powerful and innovative Knowledge Discovery framework. Fig. 3.8 depicts the three modules composing the block:

- *User Profiling*. This module takes in input the *Borè* data flow, that contains users' browsing data (i.e., which contents each user is requesting). The aim of this module is to store this information in tuples $\langle u, c, t \rangle$, where u is an identifier for the user, c identifies the content and t is the timestamp in which u accessed c . Obviously additional information (e.g., user's profile or content's structure) can be retrieved asynchronously on-demand, simply accessing the corresponding *Borè* data cartridge. Temporal information is needed to retrieve the chronological order of users' actions: in web navigation logs, in fact, data can be "naturally" interpreted as sequences; as we will study in Chapter 4, sequential data may express causality and dependency and a sequential approach allows more accurate recommendations to end user;
- *Model Inference*. It is the core module of the block. It takes in input the sequential browsing data from the *User Profiling* module and it constructs corresponding model(s). These models are extremely innovative and they produce more accurate recommendations than the classical approaches. Indeed, we extended the classical probabilistic topic models (see Chapter 4), that are widely used in different contexts to uncover the hidden structure in large text corpora. One of the main (and perhaps strong) assumption of these models is that generative process follows a bag-of-words assumption, i.e each token/word/content is independent from previous ones. More specifically, we extended the popular Latent Dirichlet Allocation model [13] by exploiting three different conditional Markovian assumptions: (i) the token generation depends on the current topic and on the previous token (*Token-Bigram Model*); (ii) the topic associated with each observation depends on topic associated with the previous one (*Topic-Bigram Model*); (iii) the token generation depends on the current and previous topic (*Token-Bitopic Model*). A detailed description of the proposed models and their performance advantages is given in Chapter 4;

⁷ <http://netflixprize.com>

- *Ranker*. This module plays his role when probabilistic topic models, corresponding to sequential browsing data, are built. It provides a personalized ranking function (strictly related to the selected model) which associates to each pair $\langle u, c \rangle$ a score. This score is used to generate the recommendation list: top- k item of the list will be suggested to the user, as contents of potential interests. These suggestions represent the output of the whole block and they are finally shown to the end user through the *Interface Composer* module of the *Borè* platform.

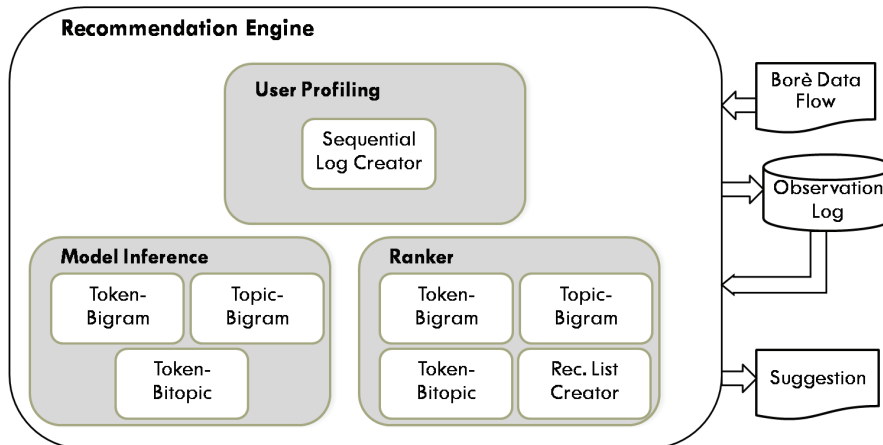


Fig. 3.8: Recommendation engine block

3.3.1 Collaborative filtering approach to recommendation

State-of-the art recommendation methods have been largely approached from a *Collaborative Filtering (CF)* perspective, which essentially consists in the analysis of past interactions between users and items, aimed at identifying suitable preference patterns in users' preference data. Tapestry [39] is the ancestor of CF systems. Recording the reactions of each user on a particular document and explicit feedback, Tapestry allows mail filtering in an area of interest basing on other people annotations.

Modern CF techniques aim of predicting the preferences values of users on given items, based on previously observed behavior. The assumption is that *users who adopted the same behavior in the past, will tend to agree also in the future*. Collaborative filtering techniques aim at finding a social neighborhood (i.e., a community) of the end user, i.e., other users with similar preferences who have experienced resources not yet known to the end user. Such resources represent potentially interesting suggestions. Once the community to which the end user belongs to is identified, those resources that are expected to be mostly interesting to him are recommended.

The main advantage in using CF techniques relies on their simplicity: only users' past ratings are used in the learning process, no further informations, like item descriptions or demographic data, are needed. CF solves some of the main drawbacks of *Content Based (CB)* approaches [62, 70]:

- CF approaches are more general and re-usable in different context, while CB techniques require the specification of a complete profile (set of features) for each user/item;
- CB techniques provide the user with a list of products whose features are “similar” to ones that he experienced in the past. This approach may imply the recommendations of redundant items and the lack of novelty;
- The effectiveness of recommendation increases as the user provides more feedback.

According to [17], collaborative filtering approaches can be classified in two classes, *Memory-based* and *Model-based*. The first class infers the preference of the active user on an item by using the database of preferences. The most common memory-based approaches are the *Nearest Neighbors* [52, 80] methods, which use the whole preference history of users and items and statistical techniques to infer similarities and then use this measures to make a prediction. Model-based approaches operate in two phases: initially the preference database is used to learn a compact model and, in the second phase, this model is used in order to infer users' preferences. Actually, memory-based approaches can rely on a model as well (i.e., similarity matrix in neighbors models), which is usually built in a off-line mode, but they still need to access to a database of preference values. Memory-based approaches are intuitive because, in the simplest case, they directly transforms stored preferences data in predictions, but they need to keep the whole dataset in memory. On the other hand, model-based approaches require less memory because they need to access only to the small-size model previously computed from the dataset, but the reason behind the provided predictions may not be easily interpretable. Moreover, memory-based approaches, such as *Neighborhood models*, are most effective at detecting *strong but local relationships*, while model-based approaches, such as *Latent Factor models*, can estimate *weak but global relationships*.

In the next of this thesis our focus is on a specific family of the model-based approaches, the Probabilistic approaches. These approaches are, in fact, the most effective way in modeling and identifying patterns within the high dimensional (and exceptionally sparse) preference data. They assume that each preference observation is randomly drawn from the joint distribution of the random variables which model users, items and preference values (if available). Probabilistic approaches will be widely discussed in next chapter, in which we will also present our extensions to classic probabilistic topic models, introducing causality and dependency through a sequential approach; proposed models provide a better framework for modeling contextual information in a recommendation scenario, when the data exhibits intrinsic temporal dependency. The effectiveness of these models is confirmed by an experimental evaluation over real-life datasets, in which we show that the sequen-

tial modeling of preference data allows more accurate recommendations in terms of precision and recall.

Probabilistic topic models for sequence data

4.1 Introduction

The design of accurate recommender systems is rapidly becoming one of the most successful application of data mining and machine learning techniques. Recommendation is a special form of information filtering, which extends the traditional concept of search, by modeling and understanding personal users' preference. The importance of an accurate Recommender System is widely witnessed by both academic and industrial efforts in the last two decades.

Probabilistic approaches are the most effective way in modeling and identifying patterns within the high dimensional (and exceptionally sparse) preference data. These approaches to recommendation assume that each preference observation is randomly drawn from the joint distribution of the random variables which model users, items and preference values (if available). In this chapter, we are going to present and discuss the application of novel probabilistic approaches for the modeling of users' preference data.

The chapter is structured as follows: Section 4.2 introduces the probabilistic approach to recommendation and describes the popular LDA model. Section 4.3 discusses the main constraint of this model, that will be relaxed in Section 4.4. Here we define sequential modeling according to different dependency assumptions, and we specify in Section 4.5 the corresponding item ranking functions for supporting recommendations. The experimental evaluation of the proposed approaches is presented in Section 4.6, in which we measure their performance in a recommendation scenario and we show that the proposed sequential modeling of preference data allows more accurate recommendations in terms of precision and recall.

4.2 Probabilistic topic models

Probabilistic approaches assume that each preference observation is randomly drawn from the joint distribution of the random variables which model users, items and preference values (if available). Typically, the random generation process follows a

“bag-of-words” assumption and preference observations are assumed to be generated independently. A key difference between probabilistic and deterministic models relies in the inference phase: while the latter approaches try to minimize directly the error made by the model, probabilistic approaches do not focus on a particular error metric; parameters are determined by maximizing the likelihood of the data, typically employing an Expectation-Maximization procedure. In addition, background knowledge can be explicitly modeled by means prior probabilities, thus allowing a direct control on overfitting within the inference procedure [53]. By modeling prior knowledge, they implicitly solve the need for regularization which affects traditional gradient-descent based latent factors approaches.

Further advantages of probabilistic models can be found in their easy interpretability: they can often be represented by using a graphical model, which summarizes the intuition behind the model by underlying causal dependencies between users, items and hidden factors. Also, they provide an unified framework for combining collaborative and content features [1, 88, 106], to produce more accurate recommendations even in the case of new users/items. Moreover, assuming that an explicit preference value is available, probabilistic models can be used to model a distribution over rating values which can be used to infer confidence intervals and to determine the confidence of the model in providing a recommendation. The underlying idea of probabilistic models based on latent factors is that each preference observation $\langle u, i \rangle$ is generated by one of k possible states, which informally model the underlying reason why u has chosen/rated i .

Different probabilistic approaches have been proposed to recommendation: *Mixture Models* [10, 64], *Probabilistic Topic Models* [11, 13] and *Probabilistic Matrix Factorization Techniques* [79]. Recent works have empirically shown that probabilistic topics models represent the state of the art in the generation of accurate personalized recommendations [5, 6].

Probabilistic Topic models [11, 13] include a suite of techniques which widely used in text analysis: they provide a low-dimensional *semantic* representation that allows the discovering of *global relationships* within data. Given a *corpus* of *documents*, the assumption behind this family of techniques is that each document may exhibit an hidden thematic structure. The intuition is that each document may exhibit multiple topics, where each topic is characterized by a probability distribution over words of a fixed size dictionary (so each word in the document is generated by a particular topic). This approach has a natural interpretation when dealing with users’ preference data: the set of users defines the corpus, each user is considered as a document, the items purchased are considered as tokens and, finally, the topics correspond, intuitively, to the reason why the users purchased particular products. A generative process for CF, based on latent topics, is shown in Fig. 4.1. This representation of the data into the latent-topic space has several advantages, as topic modeling techniques have been applied to different contexts. Example scenarios range from traditional problems (such as dimensionality reduction and classification) to novel areas (such as the generation of personalized recommendations).

In the CF context, topics could be interpreted as genres, item categories or user’s attitudes, although no prior meaning is generally associated with them. A proper

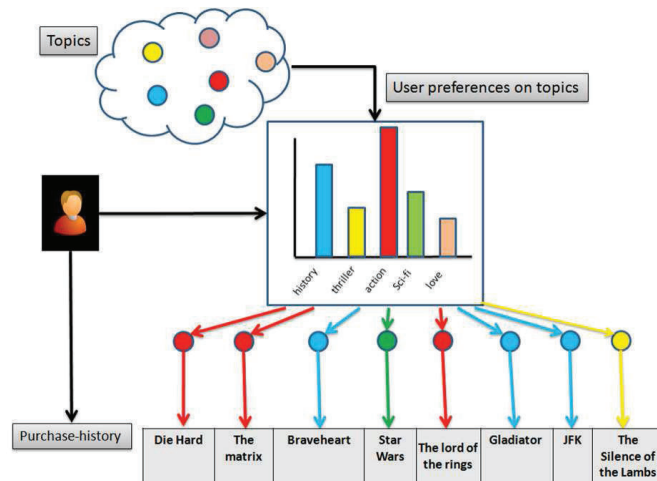


Fig. 4.1: Latent class model for CF – Generative process

definition of topics might be obtained by considering them as “abstract preference pattern”: users, or items, participate in each preference pattern with a certain degree, and these membership weights project each user/item into the latent factor space. We assume that there are a fixed number of topics, and each user is characterized by his own preference on genres. For example, in Fig. 4.1, the considered user shows a particular interest in action and historic movies, while his interest in romance is low. Each genres specifies the probability of observing each single item. Movies like “The Matrix” and “Die hard” will have a higher probability of being observed given the “action” topic, than in the context of “romance”. Given an user and his preferences on topics (which defines preferences on movies), the corresponding purchasing history can be generated by choosing a topic, and then drawing an item from the corresponding distribution over items. In the example, the first topic to be chosen is “action”, which generates the movie “Die Hard”; the process of topic and item selection is iteratively repeated to generate the complete purchase history of the current user.

The most popular probabilistic topic model is the *Latent Dirichlet Allocation* (LDA, [13]), that is a model closely linked to the probabilistic latent semantic analysis (PLSA, [55, 56]). PLSA model specifies a co-occurrence data model in which the user u and item i are conditionally independent given the state of the latent factor Z . PLSA model associates a latent variable with every observation. The main drawback of the PLSA approach is that it cannot directly model new users, because the distribution of topics w.r.t. users are specified only for those users in the training set. LDA is designed to overcome this drawback, by introducing Dirichlet priors, which provide a full generative semantic at user level and avoid overfitting. The generative process that characterized LDA can be formalized as follows:

1. For each trace $d \in \{1, \dots, M\}$ sample the topic-mixture components $\theta_d \sim \text{Dirichlet}(\alpha)$ and sequence length $n_d \sim \text{Poisson}(\xi)$
2. For each topic $k \in 1, \dots, K$
 - a) Sample token selection components $\phi_k \sim \text{Dirichlet}(\beta_k)$
3. For each trace $d \in \{1, \dots, M\}$ and $j \in \{1, \dots, N_d\}$
 - a) sample a topic $z_{d,j} \sim \text{Discrete}(\theta_d)$
 - b) sample a token $w_{d,j} \sim \text{Discrete}(\phi_{z_{d,j}})$

Where θ and ϕ represent, respectively, the per-document topic proportion and the per-corpus topic distribution, while α/β are the hyper parameters for the topic/token Dirichlet distributions. The graphical representation of the LDA-model is reported in Fig. 4.2(a).

The idea behind the model presented so far, is the starting point of more advanced approaches that include side information to achieve better results in prediction accuracy and provide tools for cold-start recommendation [1, 12, 88].

4.3 Beyond the “bag-of-words” assumption

Traditional LDA-based approaches propose a data generation process that is based on a “bag-of-words” assumption: the order of the items in a document can be neglected. This assumption fits textual data, where probabilistic topic models are able to detect recurrent co-occurrence patterns, which are used to define the topic space. However, there are several real-world applications where data can be “naturally” interpreted as sequences, such as biological data, web navigation logs, customer purchase history, etc. Ignoring the intrinsic sequentiality of the data, may result in poor modeling: according to the bag-of-word assumption, co-occurrences are modeled independently for each word, via a probability distribution over the dictionary in which some words exhibit a higher likelihood to appear than others. On the other hand, sequential data may express causality and dependency, and different topics can be used to characterize different dependency likelihoods. The focus here is the *context* where a current user acts and expresses preferences (i.e., the environment), characterized by side information, where the observations hold. Our claim is that the context can be enriched by the sequential information, and the latter allows a more refined modeling. In practice, a sequence expresses a context which provides valuable information for the modeling.

The above observation is particularly noteworthy when data express preferences made by users, and the ultimate objective is to model a user’s behavior in order to provide accurate recommendations. The analysis of the sequential patterns has important applications in modern recommender systems, which are significantly focusing on an accurate balance between personalization and contextualization techniques. For example, in Internet based streaming services for music or video (such

as Last.fm¹ and Videlectures.net²), the context of the user interaction with the system can easily be interpreted by analyzing the content previously requested. The assumption here is that the current item (and/or its genre) influences the next choice of the user. In particular, if a specific user is in the “mood” for classical music (as observed in the current choice), it is unlikely that the immediate subsequent choice will depart from the aforementioned mood, in favor of a song of different genre. Being able to capture such properties and exploiting them in recommendation strategy can greatly improve the accuracy of the recommendation. This is, obviously, also the *Caldera* scenario, in which current web content influences next user’s choice: user’s browsing history, in fact, can be “naturally” interpreted as a sequence.

Following the research direction outlined above, in this chapter we study the effects of “contextual” information in probabilistic modeling of preference data. We focus on the case where the context can be inferred from the analysis of the sequence data, and we propose some topic models which explicitly make use of dependency information. As a matter of fact, the issue has been dealt with in similar works (like, e.g., [99]). Here, we summarize and extend the approaches in the literature, by covering different ways of modeling dependency within preference data. Furthermore, we concentrate on the effects of such modeling on recommendation accuracy, as it explicitly reflects accurate modeling of user behavior.

4.4 Modeling sequence data

In a general setting, we consider a set $I = \{1, \dots, N\}$ of tokens, representing the vocabulary of possible events that can be observed. Example events are words that can be observed in a document, or items that can be purchased by a customer. A corpus $W = \{\mathbf{w}_1, \dots, \mathbf{w}_M\}$ is a collection of traces, where $\mathbf{w}_d = [w_{d,1} \cdot w_{d,2} \cdot \dots \cdot w_{d,N_d-1} \cdot w_{d,N_d}]$ is the sequence of tokens for trace d , and $w_{d,j} \in I$. The set $I_d \subseteq I$ denotes all the tokens in \mathbf{w}_d . We also assume that each token is characterized by a latent factor, called topic, triggering the underlying event. That is, a topic set $Z = \{\mathbf{z}_1, \dots, \mathbf{z}_M\}$ is associated to the data, where, again $\mathbf{z}_d = [z_{d,1} \cdot z_{d,2} \cdot \dots \cdot z_{d,N_d-1} \cdot z_{d,N_d}]$ is a latent topic sequence, and $z_{d,j} \in \{1, \dots, K\}$ is the latent topic associated with token $w_{d,j}$. By assuming that Φ and Θ are the distribution functions governing the likelihood of W and Z (with respective priors β and α), we can express the complete likelihood as:

$$P(W, Z, \Theta, \Phi | \alpha, \beta) = P(W|Z, \Phi)P(\Phi|\beta)P(Z|\Theta)P(\Theta|\alpha)$$

$$P(W|Z, \Phi) = \prod_{d=1}^M P(\mathbf{w}_d | \mathbf{z}_d, \Phi) \quad P(Z|\Theta) = \prod_{d=1}^M P(\mathbf{z}_d | \theta_d) \quad (4.1)$$

where $P(\Phi|\beta)$ and $P(\Theta|\alpha)$ are specified according to the modeling assumptions. In particular, in the standard LDA setting where all tokens are independent and exchangeable, we have:

¹ <http://last.fm>

² <http://videlectures.net>

notation	description
M	# Traces
N	# Distinct tokens
K	# Topics
W	Collection of traces, $W = \{\mathbf{w}_1, \dots, \mathbf{w}_M\}$
N_d	# tokens in trace d
\mathbf{w}_d	Token trace d , $\mathbf{w}_d = \{w_{d,1}, w_{d,2}, \dots, w_{d,N_d-1}, w_{d,N_d}\}$
$w_{d,j}$	j -th token in trace d
Z	Collection of topic traces, $Z = \{\mathbf{z}_1, \dots, \mathbf{z}_M\}$
\mathbf{z}_d	Topics for trace d , $\mathbf{z}_d = \{z_{d,1}, z_{d,2}, \dots, z_{d,N_d-1}, z_{d,N_d}\}$
$z_{d,j}$	j -th topic in trace d
$n_{d,s}^k$	number of times token s has been associated with topic k for trace d
$\mathbf{n}_{d,(\cdot)}$	vector $\mathbf{n}_{d,(\cdot)} = \{n_{d,(\cdot),1}^1, \dots, n_{d,(\cdot),N}^K\}$
$n_{d,(\cdot)}^k$	number of times topic k has been associated with trace d in the whole data
$\mathbf{n}_{(\cdot),r}^k$	vector $\mathbf{n}_{(\cdot),r}^k = \{n_{(\cdot),r,1}^k, \dots, n_{(\cdot),r,N}^k\}$
$n_{(\cdot),rs}^k$	number of times topic k has been associated with the token pair r,s in the whole data
$\mathbf{n}_{(\cdot)}^k$	vector $\mathbf{n}_{(\cdot)}^k = \{n_{(\cdot),1}^k, \dots, n_{(\cdot),N}^k\}$
$n_{(\cdot),s}^k$	number of times token s has been associated with topic k in the whole data
$\mathbf{n}_{d,(\cdot)}^{h,k}$	vector $\mathbf{n}_{d,(\cdot)}^{h,k} = \{n_{d,(\cdot),1}^{h,k}, \dots, n_{d,(\cdot),N}^{h,k}\}$
$n_{d,(\cdot)}^{h,k}$	number of times that topic pair h,k has been associated with the trace d
$n_{d,(\cdot)}^{h,(\cdot)}$	number of times that a topic pair, that begins with topic h , has been associated with the trace d
$\mathbf{n}_{(\cdot)}^{h,k}$	vector $\mathbf{n}_{(\cdot)}^{h,k} = \{n_{(\cdot),1}^{h,k}, \dots, n_{(\cdot),N}^{h,k}\}$
$n_{(\cdot),s}^{h,k}$	number of times that topic pair h,k has been associated with the token s in the whole data
α	(LDA, TokenBigram and TokenBitopic Model) hyper parameters for topic Dirichlet distribution $\alpha = \{\alpha_1, \dots, \alpha_K\}$ (Topic Bigram Model) set of hyper parameters for topic Dirichlet distribution $\alpha = \{\alpha_0, \dots, \alpha_K\}$
α_h	hyper parameters for topic Dirichlet distribution $\alpha_h = \{\alpha_{h,1}, \dots, \alpha_{h,K}\}$
β	(LDA and TopicBigram Model) set of hyper parameters for token Dirichlet distribution $\beta = \{\beta_1, \dots, \beta_K\}$ (TokenBigram Model) set of hyper parameters for token Dirichlet distribution $\beta = \{\beta_{1,1}, \dots, \beta_{K,1}, \dots, \beta_{1,2}, \dots, \beta_{K,2}, \dots, \beta_{K,N}\}$ (TokenBitopic Model) set of hyper parameters for token Dirichlet distribution $\beta = \{\beta_{1,1}, \dots, \beta_{K,1}, \dots, \beta_{1,2}, \dots, \beta_{K,2}, \dots, \beta_{K,K}\}$
β_k	hyper parameters for token Dirichlet distribution $\beta_k = \{\beta_{k,1}, \dots, \beta_{k,N}\}$
$\beta_{k,s}$	hyper parameters for token Dirichlet distribution $\beta_{k,s} = \{\beta_{k,s,1}, \dots, \beta_{k,s,N}\}$
$\beta_{h,k}$	hyper parameters for token Dirichlet distribution $\beta_{h,k} = \{\beta_{h,k,1}, \dots, \beta_{h,k,N}\}$
Θ	matrix of parameters θ_d
θ_d	mixing proportion of topics for trace d
$\vartheta_{d,k}$	mixing coefficient of the topic k for trace d
$\vartheta_{d,h,k}$	mixing coefficient of the topic sequence h,k for the trace d
Φ	(LDA and TopicBigram Model) matrix of parameters $\varphi_k = \{\varphi_{k,s}\}$ (TokenBigram Model) matrix of parameters $\varphi_k = \{\varphi_{k,r,s}\}$ (TokenBitopic Model) matrix of parameters $\varphi_{h,k} = \{\varphi_{h,k,s}\}$
$\varphi_{k,s}$	mixing coefficient of the topic k for the token s
$\varphi_{k,r,s}$	mixing coefficient of the topic k for the token sequence r,s
$\varphi_{h,k,s}$	mixing coefficient of the topic sequence h,k for the token s
$\mathbf{Z}_{-(d,j)}$	$\mathbf{Z} - \{z_{d,j}\}$
$\Delta(\mathbf{q})$	Dirichlet's Delta $\Delta(\mathbf{q}) = \frac{\prod_{p=1}^P \Gamma(q_p)}{\Gamma(\sum_{p=1}^P \Gamma(q_p))}$

Table 4.1: Summary of the notation used

$$\begin{aligned}
P(\mathbf{w}_d|\mathbf{z}_d, \Phi) &= \prod_{j=1}^{N_d} P(w_{d,j}|z_{d,j}, \Phi) & P(w|k, \Phi) &= \prod_{s=1}^N \phi_{k,s}^{\delta_{s,w}} \\
P(\mathbf{z}_d|\theta_d) &= \prod_{j=1}^{N_d} P(z_{d,j}|\theta_d) & P(z|\theta_d) &= \prod_{k=1}^K \vartheta_{d,k}^{\delta_{k,z}} \\
P(\Theta|\alpha) &= \prod_{d=1}^M P(\theta_d|\alpha) & P(\theta_d|\alpha) &= \frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K \vartheta_{d,k}^{\alpha_k-1} \\
P(\Phi|\beta) &= \prod_{k=1}^K P(\phi_k|\beta_k) & P(\phi_k|\beta_k) &= \frac{\Gamma(\sum_{s=1}^N \beta_{k,s})}{\prod_{s=1}^N \Gamma(\beta_{k,s})} \prod_{s=1}^N \phi_{k,s}^{\beta_{k,s}-1}
\end{aligned} \tag{4.2}$$

Here, $\delta_{a,b}$ represents the Kronecker delta function, returning 1 when $a = b$ and 0 otherwise. Fig. 4.2(a) graphically describes the generative process. The joint topic-data probability can be obtained by marginalizing over the Φ and Θ components:

$$P(W, Z|\alpha, \beta) = \int_{\Phi} \int_{\Theta} P(W|Z, \Phi) P(\Phi|\beta) P(Z|\Theta) P(\Theta|\alpha) d\Phi d\Theta$$

In the following, we model further assumptions on both \mathbf{w}_d and \mathbf{z}_d , which explicitly reject the exchangeability assumption and instead rely on the idea of sequential dependency. We concentrate on three basic models, which in a sense subsume the core of sequential modeling. Here, a sequence can be modeled as a stationary first order Markov chain:

- A Markovian process naturally models the sequential nature of the data, where dependencies among past and future tokens reflect changes over time that are still governed by similar features;
- the chain is stationary, as a fixed number of tokens is likely to frequently appear in sequences;
- the order of the chain is 1 because the possibility that two subsequent tokens share some features is more likely than that of two tokens distant in time³.

We analyze each proposed model in turn.

Token-Bigram model

In this model, we assume that \mathbf{w}_d represents a first-order Markov chain, where, each token $w_{d,j}$ depends on the most recent token $w_{d,j-1}$ observed by far. This is essentially the same model proposed in [19, 99], and the probability of a trace has to be changed from eq. 4.2 as

$$P(\mathbf{w}_d|\mathbf{z}_d, \Phi) = \prod_{j=1}^{N_d} P(w_{d,j}|w_{d,j-1}, z_{d,j}, \Phi) \tag{4.3}$$

³ It is also worth noticing that higher order dependencies introduce an unpractical computational overhead, as the number of parameters grows exponentially with the order of the chain [10].

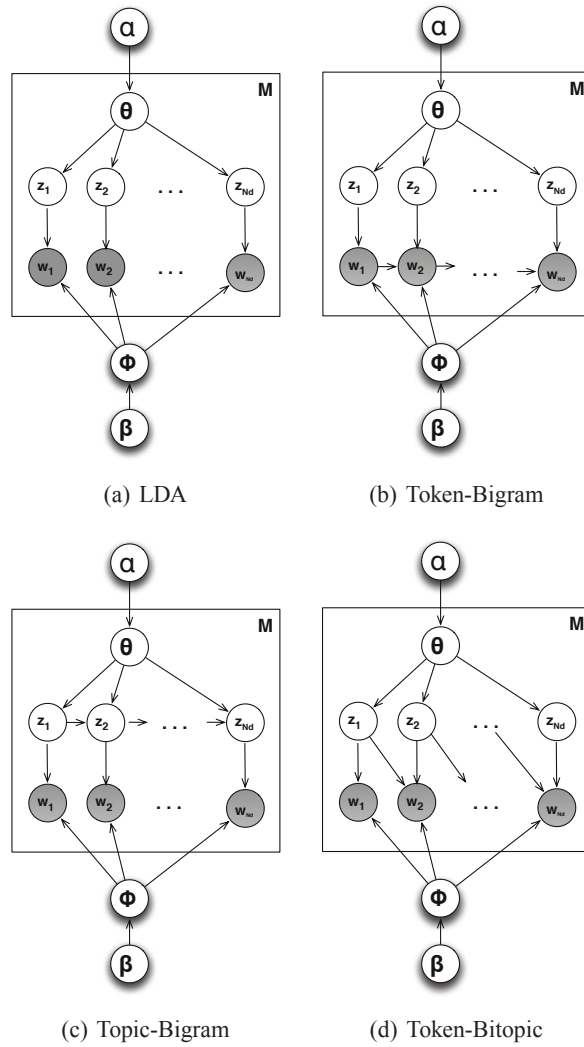


Fig. 4.2: Graphical models

In practice, a token $w_{d,j}$ is generated according to a multinomial distribution $\phi_{z_{d,j}, w_{d,j-1}}$ which depends on both the current topic $z_{d,j}$ and the previous token $w_{d,j-1}$. (Notice that when $j = 1$, the previous token is empty and the multinomial resolves to $\phi_{z_{d,j}}$, representing the initial status of a Markov chain). The conjugate prior for ϕ can be defined as:

$$P(\Phi|\beta) = \prod_{k=1}^K \prod_{r=0}^N P(\phi_{k,r}|\beta_{k,r}) = \prod_{k=1}^K \prod_{r=0}^N \frac{\Gamma(\sum_{s=1}^N \beta_{k,r,s})}{\prod_{s=1}^N \Gamma(\beta_{k,r,s})} \prod_{s=1}^N \phi_{k,r,s}^{\beta_{k,r,s}-1}$$

Since the Markovian process does not affect the topic sampling, both $P(\mathbf{z}_d|\theta_d)$ and $P(\Theta|\alpha)$ are defined as in eq. 4.2. The generative model, depicted in Fig. 4.2(b), can be described as follows:

1. For each trace $d \in \{1, \dots, M\}$ sample the topic-mixture components $\theta_d \sim \text{Dirichlet}(\alpha)$ and sequence length $n_d \sim \text{Poisson}(\xi)$
2. For each topic $k \in 1, \dots, K$ and token $r \in \{0, \dots, N\}$
 - a) Sample token selection components $\phi_{k,r} \sim \text{Dirichlet}(\beta_{k,r})$
3. For each trace $d \in \{1, \dots, M\}$ and $j \in \{1, \dots, N_d\}$
 - a) sample a topic $z_{d,j} \sim \text{Discrete}(\theta_d)$
 - b) sample a token $w_{d,j} \sim \text{Discrete}(\phi_{z_{d,j}, w_{d,j-1}})$

Notice that we explicitly assume the existence of a family $\{\beta_{k,r}\}$ with $k = \{1, \dots, K\}$ and $r = \{0, \dots, N\}$ of Dirichlet coefficients, and of a special token $r = 0$ which represents the previous token of the first token of each trace. As shown in [99], different modeling strategies (e.g., shared priors $\beta_{k,r,s} = \beta_s$) can affect the accuracy of the model.

By algebraic manipulations, the joint token-topic distribution can be simplified into:

$$P(W, Z|\alpha, \beta) = \left(\prod_{d=1}^M \frac{\Delta(\mathbf{n}_{d,(\cdot)} + \alpha)}{\Delta(\alpha)} \right) \left(\prod_{k=1}^K \prod_{r=0}^N \frac{\Delta(\mathbf{n}_{(\cdot),r}^k + \beta_{k,r})}{\Delta(\beta_{k,r})} \right) \quad (4.4)$$

The latter is the basis for developing a stochastic EM strategy [10, section 11.1.6], where the E step consists in a collapsed Gibbs sampling procedure [10, 49] for estimating \mathbf{Z} , and the M step estimates both the predictive distributions Θ and Φ and the hyper parameters α and β given \mathbf{Z} . Within Gibbs sampling, topics are iteratively sampled, according to the probability:

$$P(z_{d,j} = k | \mathbf{Z}_{-(d,j)}, \mathbf{W}) \propto \left(n_{d,(\cdot)}^k + \alpha_k - 1 \right) \cdot \frac{n_{(\cdot),r,s}^k + \beta_{k,r,s} - 1}{\sum_{s'=1}^N n_{(\cdot),r,s'}^k + \beta_{k,r,s'} - 1} \quad (4.5)$$

relative to the topic to associate with the n -th token of the d -th trace, where $w_{d,j-1} = r$ and $w_{d,j} = s$.

Given \mathbf{Z} , the parameters Θ and Φ can be estimated according to the following equations:

$$\vartheta_{d,k} = \frac{n_{d,(\cdot)}^k + \alpha_k}{\sum_{k'=1}^K (n_{d,(\cdot)}^{k'} + \alpha_{k'})} \quad \varphi_{k,r,s} = \frac{n_{(\cdot),r,s}^k + \beta_{k,r,s}}{\sum_{s'=1}^N (n_{(\cdot),r,s'}^k + \beta_{k,r,s'})} \quad (4.6)$$

The estimation of the hyper parameters will be approached later in the chapter.

Topic-Bigram model

A different approach can be taken by assuming that sequentiality regards topics, rather than tokens. That is, we can still consider tokens independent to each other and related to a latent topic. However, since topics represent the ultimate factors underlying a token appearance in the sequence, correlation between topics can better model an evolution of the underlying themes. Assuming a first-order Markovian dependency, the probability of a sequence of latent topics in eq. 4.2 can be redefined as:

$$P(\mathbf{z}_d|\theta_d) = \prod_{j=1}^{N_d} P(z_{d,j}|z_{d,j-1}, \theta_d) \quad (4.7)$$

The difference here is in the distribution generating $z_{d,j}$, which is a multinomial $\theta_{d,z_{d,j-1}}$ parametrized by both a trace d and a previously sampled topic $z_{d,j-1}$. The conjugate Dirichlet distributions can be expressed as:

$$P(\Theta|\alpha) = \prod_{d=1}^M \prod_{h=0}^K \frac{\Gamma(\sum_{k=1}^K \alpha_{h,k})}{\prod_{k=1}^K \Gamma(\alpha_{h,k})} \prod_{k=1}^K \vartheta_{d,h,k}^{\alpha_{h,k}-1} \quad (4.8)$$

$P(\mathbf{w}_d|\mathbf{z}_d, \Phi)$ and $P(\Phi|\beta)$ are still defined as in eq. 4.2. Again, the generative process is shown in Fig. 4.2(c) and described below.

1. For each trace $d \in \{1, \dots, M\}$ and topic $h \in \{0, \dots, K\}$ sample topic-mixture components $\theta_{d,h} \sim \text{Dirichlet}(\alpha_h)$ and sequence length $N_d \sim \text{Poisson}(\xi)$
2. For each topic $k = 1, \dots, K$
 - a) Sample token selection components $\phi_k \sim \text{Dirichlet}(\beta_k)$
3. For each $d \in \{1, \dots, M\}$ and $j \in \{1, \dots, N_d\}$ sequentially,
 - a) sample a topic $z_{d,j} \sim \text{Discrete}(\theta_{d,z_{d,j-1}})$
 - b) sample a token $w_{d,j} \sim \text{Discrete}(\phi_{z_{d,j}})$

Here, $h = 0$ is a special topic that precedes the first topic of each trace.

The joint token-topic distribution becomes:

$$P(W, Z|\alpha, \beta) = \left(\prod_{d=1}^M \prod_{h=0}^K \frac{\Delta(\mathbf{n}_{d,(\cdot)}^k + \alpha_h)}{\Delta(\alpha_h)} \right) \left(\prod_{k=1}^K \frac{\Delta(\mathbf{n}_{(\cdot)}^k + \beta_k)}{\Delta(\beta_k)} \right) \quad (4.9)$$

and the corresponding collapsed Gibbs sampler works by iteratively sampling a topic k relative to token $w_{d,j} = s$ of trace d according to the following:

$$P(z_{d,j} = k | \mathbf{Z}_{-(d,j)}, \mathbf{W}) \propto \frac{n_{d,(\cdot)}^{z_{d,j-1},k} + \alpha_{z_{d,j-1},k} - 1}{\sum_{k'} n_{d,(\cdot)}^{z_{d,j-1},k'} + \alpha_{z_{d,j-1},k'} - 1} \cdot \frac{n_{d,(\cdot)}^{k,z_{d,j+1}} + \alpha_{k,z_{d,j+1}} - 1}{\sum_{k'} n_{d,(\cdot)}^{k',z_{d,j+1}} + \alpha_{k',z_{d,j+1}} - 1} \cdot \frac{n_{(\cdot),s}^k + \beta_{k,s} - 1}{\sum_{s'=1}^N n_{(\cdot),s'}^k + \beta_{k,s'} - 1} \quad (4.10)$$

Also, the multinomial parameters can be estimated according to the following equations:

$$\vartheta_{d,h,k} = \frac{n_{d,(\cdot)}^{h,k} + \alpha_{h,k}}{\sum_{k'=1}^K n_{d,(\cdot)}^{h,k'} + \alpha_{h,k'}} \quad \Phi_{k,s} = \frac{n_{(\cdot),s}^k + \beta_{k,s}}{\sum_{s'=1}^N n_{(\cdot),s'}^k + \beta_{k,s'}} \quad (4.11)$$

Token-Bitopic model

In the last model, we still relate tokens to past events. However, the events we are interested in are the recent latent topics which trigger the past tokens. The generative model is shown in Fig. 4.2(d). Again, topic selection probability is defined like in eq. 4.2, whereas token selection probability can be defined in terms of the multinomial $\phi_{z_{d,j}, z_{d,j-1}}$ (and its related conjugate):

$$P(\mathbf{w}_d | \mathbf{z}_d, \Phi) = \prod_{j=1}^{N_d} P(w_{d,j} | z_{d,j}, z_{d,j-1}, \Phi) \quad (4.12)$$

$$P(\Phi | \beta) = \prod_{h=0}^K \prod_{k=1}^K \frac{\Gamma(\sum_{s=1}^N \beta_{h,k,s})}{\prod_{s=1}^N \Gamma(\beta_{h,k,s})} \prod_{s=1}^N \phi_{h,k,s}^{\beta_{h,k,s}-1} \quad (4.13)$$

These assumptions are at the basis of the following generative process.

1. For each trace $d \in \{1, \dots, M\}$ sample topic-mixture components $\theta_d \sim \text{Dirichlet}(\alpha)$ and sequence length $N_d \sim \text{Poisson}(\xi)$
2. For each topic pair h,k , where $h \in \{0, \dots, K\}$ and $k \in \{1, \dots, K\}$
 - a) Sample token selection components $\phi_{h,k} \sim \text{Dirichlet}(\beta_{h,k})$
3. For each $d \in \{1, \dots, M\}$ and $j \in \{1, \dots, N_d\}$ in sequence:
 - a) sample a topic $z_{d,j} \sim \text{Discrete}(\theta_d)$
 - b) sample a token $w_{d,j} \sim \text{Discrete}(\phi_{z_{d,j}, z_{d,j-1}})$

Once again $h = 0$ is the special topic which precedes all the first topics of the traces. As usual, by algebraic manipulations, the joint token-topic distribution can be expressed as

$$P(W, Z | \alpha, \beta) = \left(\prod_{d=1}^M \frac{\Delta(\mathbf{n}_{d,(\cdot)} + \alpha)}{\Delta(\alpha)} \right) \left(\prod_{h=0}^K \prod_{k=1}^K \frac{\Delta(\mathbf{n}_{(\cdot)}^{h,k} + \beta_{h,k})}{\Delta(\beta_{h,k})} \right) \quad (4.14)$$

which induce the following inference steps:

E step: for the token $w_{d,j} = s$ at position j in trace d , sample a topic k according to the following probability:

$$P(z_{d,j} = k | \mathbf{Z}_{-(d,j)}, \mathbf{W}) \propto \left(n_{d,(\cdot)}^k + \alpha_k - 1 \right) \cdot \frac{n_{(\cdot),s}^{z_{d,j-1},k} + \beta_{z_{d,j-1},k,s} - 1}{\sum_{s'=1}^N n_{(\cdot),s'}^{z_{d,j-1},k} + \beta_{z_{d,j-1},k,s'} - 1} \cdot \frac{n_{(\cdot),s}^{k,z_{d,j+1}} + \beta_{k,z_{d,j+1},s} - 1}{\sum_{s'=1}^N n_{(\cdot),s'}^{k,z_{d,j+1}} + \beta_{k,z_{d,j+1},s'} - 1} \quad (4.15)$$

M Step: estimate multinomial probabilities according to the following equations:

$$\vartheta_{d,k} = \frac{n_{d,(\cdot)}^k + \alpha_k}{\sum_{k'=1}^K n_{d,(\cdot)}^{k'} + \alpha_{k'}} \quad \varphi_{h,k,s} = \frac{n_{(\cdot),s}^{h,k} + \beta_{h,k,s}}{\sum_{s'=1}^N n_{(\cdot),s'}^{h,k} + \beta_{h,k,s'}} \quad (4.16)$$

4.4.1 Log-likelihoods

A crucial component in the inference and estimation steps is the computation of the data likelihood. In general, the likelihood function is defined as:

$$\begin{aligned} P(\mathbf{W}) &= \prod_{d=1}^M P(\mathbf{w}_d) = \prod_{d=1}^M P(w_{d,1} \cdots w_{d,N_d}) \\ &= \prod_{d=1}^M \sum_{k=1}^K P(w_{d,1} \cdots w_{d,N_d}, z_{d,N_d} = k) \end{aligned}$$

Now, each model differs in the way the $P(w_{d,1} \cdots w_{d,N_d}, z_{d,N_d})$ component is defined.

Token-Bigram

Bayes rule and the first order Markov assumption over tokens simplifies the above probability into:

$$\log P(\mathbf{W}) = \sum_{d=1}^M \log \left(\prod_{j=1}^{N_d} \sum_k \vartheta_{d,k} \varphi_{k,w_{d,j-1},w_{d,j}} \right) \quad (4.17)$$

Topic-Bigram

By algebraic manipulations (see [10, section 13.2] for details), we obtain

$$\begin{aligned} P(w_{d,1} \cdots w_{d,N_d}, z_{d,N_d} = k) &= P(w_{d,1} \cdots w_{d,N_d} | z_{d,N_d} = k) P(z_{d,N_d} = k) \\ &= \varphi_{k,w_{d,N_d}} \sum_h P(w_{d,1} \cdots w_{d,N_d-1}, z_{d,N_d-1} = h) \vartheta_{d,h,k} \end{aligned}$$

The result is a recursive equation which can be simplified into the following γ function:

$$\gamma_k(\mathbf{w}_d; 1) = \varphi_{k,w_{d,1}}; \quad \gamma_k(\mathbf{w}_d; j) = \varphi_{k,w_{d,j}} \sum_h \gamma_h(\mathbf{w}_d; j-1) \vartheta_{d,h,k}$$

Substituting into the likelihood, yields:

$$\log P(\mathbf{W}) = \sum_{d=1}^M \log \left(\sum_k \gamma_k(\mathbf{w}_d; N_d) \right) \quad (4.18)$$

Token-Bitopic

The term $P(w_{d,1} \cdots w_{d,N_d} | z_{d,N_d} = k)$ can be decomposed according to the assumption of independence among topics:

$$\begin{aligned} P(w_{d,1}, \dots, w_{d,N_d} | z_{d,N_d} = k) \\ &= \sum_{h=1}^K \vartheta_{d,h} P(w_{d,1} \cdots w_{d,N_d} | z_{d,N_d-1} = h, z_{d,N_d} = k) \\ &= \sum_{h=1}^K \vartheta_{d,h} \phi_{h,k,s} P(w_{d,1} \cdots w_{d,N_d-1} | z_{d,N_d-1} = h) \end{aligned}$$

where $w_{d,N_d} = s$. Again, the latter yields the following recursive equations

$$\gamma_k(\mathbf{w}_d, 1) = \phi_{w_{d,1}, \varepsilon, k}; \quad \gamma_k(\mathbf{w}_d, j) = \sum_h \gamma_h(\mathbf{w}_d, j-1) \vartheta_{d,h} \phi_{w_{d,j}, h, k}$$

where ε is a special topic, referring to the begin of the trace. The likelihood can hence be expressed as:

$$\log P(\mathbf{W}) = \sum_{d=1}^M \log \left(\sum_k \gamma_k(\mathbf{w}_d; N_d) \vartheta_{d,k} \right) \quad (4.19)$$

4.4.2 Estimating the hyper parameters

We consider asymmetric Dirichlet priors over the trace topic distributions and a symmetric prior over the topic distributions. This modeling strategy has been reported to achieve important advantages over the symmetric version [100]. For the token-bigram and token-bitopic models, we adopted the procedure for updating the prior α as described in [49, 67]. The topic-bigram model requires a difference formulation of the latter. Given a state of the Markov chain Z , the optimal α -hyper parameters can be computed by maximizing the likelihood of the observed pseudo-counts $n_{d,(\cdot)}^{h,k}$ via the fixed-point iteration method:

$$\alpha_{h,k}^{new} = \alpha_{h,k} \frac{\sum_{d=1}^M \Psi(n_{d,(\cdot)}^{h,k} + \alpha_{h,k}) - M \Psi(\alpha_{h,k})}{\sum_{d=1}^M \Psi(n_{d,(\cdot)}^{h,(\cdot)} + \sum_{k'=1}^K \alpha_{h,k'}) - M \Psi(\sum_{k'=1}^K \alpha_{h,k'})} \quad (4.20)$$

where $\Psi(\cdot)$ indicates the digamma function.

4.5 Application to recommender systems

The general framework introduced above, has a natural interpretation when dealing with users' preference data: the set of users defines the corpus, each user is considered as a trace, the items purchased are considered as tokens and, finally, the topics correspond, intuitively, to the reason why the users purchased particular products. In the following, we assume that a user can be denoted by a unique index d , and a previous history is given by \mathbf{w}_d of size N_d . We are interested in providing a ranking for s , the $N_d + 1$ -th choice w_{d,N_d+1} .

LDA

Following [5] we adopt the following ranking function:

$$\text{rank}(s, d) = \sum_{k=1}^K P(s|z_{d, N_d+1} = k) P(z_{d, N_d+1} = k | \theta_d) = \sum_{k=1}^K \phi_{k, s} \cdot \vartheta_{d, k}$$

It has been shown [5] that LDA, equipped with the above ranking function, significantly outperforms the most significant approaches to modeling user preferences. Hence, it is a natural baseline function upon which to measure the performance of the other approaches proposed in this chapter.

Token-Bigram

The dependency of the current selection from the previous history can be made explicit, thus yielding the following upgrade to the LDA ranking function:

$$\text{rank}(s, d) = \sum_{k=1}^K P(s|z_{u, N_d+1} = k, \mathbf{w}_d) P(z_{d, N_d+1} = k | \theta_d) = \sum_{k=1}^K \phi_{k, r, s} \cdot \vartheta_{d, k}$$

where $r = w_{u, N_d}$ is the last item selected by user d in her currently history.

Topic-Bigram

This situation resembles the forward-backward algorithm for the hidden Markov models [10]. In practice, we need to build a recursive chain of probabilities, representing a hypothetical random walk among the hidden topics. As above, we can define the following rank:

$$\begin{aligned} \text{rank}(s, d) &= \sum_{k=1}^K P(w_{d, N_d+1} = s, z_{u, N_d+1} = k | \mathbf{w}_d) \\ &= \sum_{k=1}^K \frac{P(w_{d, 1} \cdots w_{d, N_d+1}, z_{d, N_d+1})}{P(\mathbf{w}_d)} \end{aligned}$$

which requires solving $P(w_{d, 1} \cdots w_{d, N_d+1}, z_{d, N_d+1})$. As shown in the previous section, the latter can be computed recursively by exploiting the γ function. Hence, the ranking function can be formulated as:

$$\text{rank}(s, d) \propto \sum_{k=1}^K \gamma_k(\mathbf{w}_d, s, N_d + 1)$$

Token-Bitopic

Since in this case item selection depends on the previous topics, by exploiting the γ function, we can define the following:

$$\begin{aligned} \text{rank}(s, d) &= P(w_{d, N_d+1} = s | \mathbf{w}_d) \propto \sum_{k=1}^K P(w_{d, 1} \cdots w_{d, N_d+1}, z_{d, N_d+1}) \\ &= \sum_{k=1}^K \gamma_k(\mathbf{w}_d, s, N_d + 1) \vartheta_{d, k} \end{aligned}$$

4.6 Experimental evaluation

In this section we study the behavior of the proposed models, compared to some baseline models. In particular, we study two main aspects.

- On a general setting, we study how the proposed method perform in terms of quality. We measure the quality as a function of the likelihood, as explained in Section 4.6.1;
- On a more specific setting, we compare the models in the envisaged recommendation application scenario. Here, the quality of a model is measured indirectly, in terms of the accuracy of the recommendations it boosts. This is explained in Section 4.6.2.

4.6.1 Perplexity

Topic models are typically evaluated by either measuring performance on some secondary task, such as document classification or information retrieval, or by estimating the probability of unseen held-out traces given some training traces. Notably, a better model will give rise to a higher probability of held-out traces, on average.

Since log likelihoods are usually large negative numbers, perplexity is used instead [13, 49], the latter being defined as the reciprocal geometrical mean of the token likelihoods in the test corpus given the data used to train the model:

$$Perp(\mathbf{W}_{Test}|\mathbf{W}_{Train}) = \exp \left\{ - \frac{\sum_{d=1}^{N_{Test}} \log P(\mathbf{w}_d|\mathbf{W}_{Train})}{\sum_{d=1}^{N_{Test}} n_d} \right\}$$

Evaluating $P(\mathbf{w}_d|\mathbf{W}_{Train})$ is a little tricky, as exact inference would require integrating over all possible model parameters. In [101] authors discuss some methods for an accurate inference using a point estimate. In our experiments we adopted the evaluation methods based on document completion. This method offers the advantage of providing unbiased estimates, as it infers the missing parameters on a separate part of the document, and then to evaluate the perplexity on the remaining part. In short, the evaluation methodology can be summarized as follows:

- for each $w_d \in \mathbf{w}^{Test}$
 1. let $w_d^{(1)}$ and $w_d^{(2)}$ be an arbitrary split of w_d .
 2. for $s = 1, \dots, S$
 - a) Sample $z^{(1,s)} \sim P(z^{(1,s)}|w_d^{(1)}, W_{train}, \alpha, \beta, \Phi)$ using the Gibbs Sampling equations;
 - b) estimate $\theta_d^{(s)}$ from $z^{(1,s)}$;
 3. Approximate $P(w_d|\mathbf{W}_{Train})$ with $\frac{1}{S} \sum_s P(w_d^{(2)}|\theta_d^{(s)}, \Phi)$, where the latter is computed by exploiting the formulas in Section 4.4.1.

Following [99], in the experiments we use a dataset composed by drawing 150 Psychological Review abstracts from the data made available by Griffith and

Steyvers⁴. The drawing was made among those documents containing at least 54 tokens. Also, we preprocessed the data as specified in [99], by remapping all numbers with the special token `NUMBER`, and all items with frequency 1 in the training set or appearing as tokens in the test set but not in the training set as `UNSEEN`. The result of the cleaning process is a vocabulary of 860 items. Starting with the cleaned dataset, we did several random splits of the dataset, by choosing 100 documents as training data, and the keeping the remaining ones as test data. The splits roughly maintained the proportion 67-33% on the tokens.

In the following we report the results obtained by the three proposed models. The results are compared with LDA. We also compare the models with the DCMLDA model [34]. The latter is a modification of LDA to account for the tendency of tokens to appear in bursts, that is if a token appears once in a trace, it is more likely to be appear again. DCMLDA does not model sequentiality, however burstiness can also be interpreted as non-independence between tokens. In this respect, it is interesting how the proposed models compare to it. It is worth noticing, however, that burstiness is not necessarily alternative to sequentiality, as the approaches proposed in this chapter can easily be adapted to model a combination of burstiness and sequentiality.

Fig. 4.3(a) reports the average perplexity on the test data. The values plot the error bars related to the perplexity values. Fig. 4.3(b) also analyzes the pairwise comparisons: each of the three methods proposed here is compared with the baselines, and the difference in perplexity (in average and standard error) is plotted.

DCMLDA exhibits the best perplexity, as a result of the customized fitting of token probabilities to a specific document. As a matter of fact, the documents we are investigating here seem to naturally comply with the burstiness assumption.

Also, TokenBitopic seems to worsen the performance as the number of topics increase. This behavior is worth further explanation. The model conditions the probability of appearance of a token to a pair of latent factors. In a sense, this makes the model comparable to a “fresh” LDA model, where the number of latent factors is quadratic in K : in practice, a TokenBitopic model with $K = 4$ can be deemed similar to an LDA model with $K = 16$ topics, and each pair of latent factors is associated to a specific latent factor in the quadratic LDA model. In Fig. 4.3(c) we compare the two models: the models show the same tendency.

For the rest, models clearly outperform LDA. However, the TokenBigram model requires further explanation. Both the sampling process and the item selection probabilities rely on the frequencies of bigrams. Zero-frequency bigrams appearing in the test set compromise the evaluation just like zero-frequency items. We chose to treat them by associating them with a default frequency. Fig. 4.3(d) shows how this affects the evaluation: Here, `NoP` corresponds to keeping the original frequency, whereas `P3` associates a frequency which implicitly corresponds to flattening all the zero-frequency bigrams to a default `UNSEEN` bigram. The latter is the one reported in

⁴ http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm

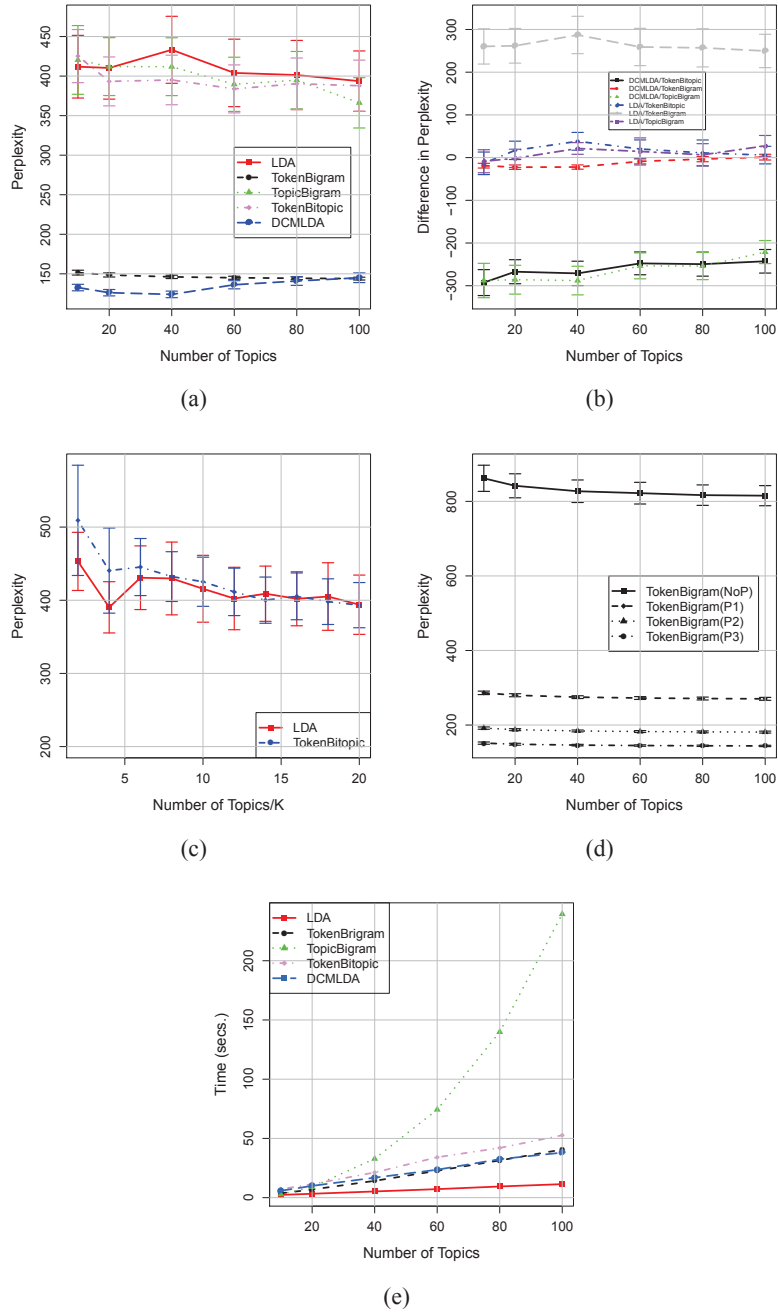


Fig. 4.3: Performance on psychreview data

Fig. 4.3(a). The approaches P1 and P2 correspond to intermediate solutions, where the default frequency of the (implicit) UNSEEN bigram is lowered⁵.

Finally, Fig. 4.3(e) denotes the running times of the training algorithms on the training data. Although the TopicBigram model requires less parameters than the TokenBitopic approach, the learning time of the first one is considerably larger. This is mainly due to the larger number of hyper parameters ($K \times K$ vs K) and to the complexity of the M step for the update of the hyper parameters α .

4.6.2 Recommendation accuracy

In this section we present an empirical evaluation of the proposed models which focuses on the recommendation problem. Given the past observed preferences of a users, the goal of a recommender systems is to provide her with personalized (and contextualized) recommendations about previously non-purchased items that meet her interest. We evaluate the proposed techniques by measuring their predictive abilities on two datasets, namely IPTV1 and IPTV2. These data were collected by analyzing the pay-per-view movies purchased by the users of two European IPTV providers over a period of several months [4,26]. The original data have been preprocessed by removing users with less than 10 purchases. We perform a chronological split of the data by selecting the final 20% purchases of each user as test data, and using the remaining data for training purposes. The main features of the datasets are summarized in table 4.2.

The two datasets exhibit a substantial difference in the frequencies of bigrams, as shown in Fig. 4.4: in particular, IPTV2 exhibits frequencies which differ of an order of magnitude. Hence, by comparing the results of the proposed algorithms, we can characterize the effects of sparsity on the performances of the proposed methods.

Testing protocol

Let \mathbf{W}_{Train} and \mathbf{W}_{Test} denote respectively training and test data. To evaluate the capabilities of the considered approaches in generating accurate recommendations, we check whether an actual token can be included into an hypothetical recommendation list containing H items, generated according to the model. More specifically the following protocol is adopted, which is justified and detailed in [5]:

- For each user u , let \mathbf{w}'_u be the trace associated to u in \mathbf{W}_{Train} , and \mathbf{w}_u the trace in \mathbf{W}_{Test} (with $n_u = |\mathbf{w}_u|$). For each token $w_{u,n} \in \mathbf{w}_u$:
 - Generate the candidate list C_u by randomly drawing c items $i \neq w_{u,n}$ such that $i \notin I_{\mathbf{w}'_u}$;
 - add $w_{u,n}$ to C_u and sort the list according to the scoring function provided by the RS;

⁵ Clearly this is where non-parametric methods should be used to provide a gradual step into the TokenBigram model. The integration of non-parametric techniques in the TokenBigram would better handle cases in which there is less data and it would automatically solve the treatment of the zero-frequency items.

	IPTV1		IPTV2	
	Training	Test	Training	Test
Users	16,237	16,153	64,334	63,878
Items	759	731	2802	2777
Evaluations	314,042	78,557	1,224,790	306,271
Avg # evals (user)	19	5	19	5
Avg # evals (item)	414	107	437	110
Min # evals (user)	4	1	4	1
Min # evals (item)	5	1	5	1
Max # evals (user)	252	15	497	17
Max # evals (item)	2284	1527	9606	3167
Avg time between two evals				
per user	13 days		6 days	
per item	9 hours		23 hours	

Table 4.2: IPTV1 and IPTV2 datasets main features

- Record the position of the $w_{u,n}$ in the ordered list: if it belongs to the top- H items, we have a *hit* otherwise, we have a *miss*.

Recall and precision relative to u can hence be defined based on the number of hits. Recall can be defined as the number of hits, relative to the expected number of relevant items (which are all the items in \mathbf{w}_u). Also, precision represents the probability that the top-ranked items are actually a hit (and hence it represents the likelihood of a hit weighted by the size H) of the recommendation list. In formulas:

$$Recall(u, H) = \frac{\#hits}{n_u} \quad Precision(u, H) = \frac{\#hits}{H \times n_u} = \frac{recall(u, H)}{H} \quad (4.21)$$

The final precision and recall values are obtained averaging on all users. All the considered models were run varying the number of topics. We perform 5000 Gibbs Sampling iterations, discarding the first 1000 (burn in period), and with a sample lag of 30. The length of the candidate random list is set to 250 for IPTV1 and 1000 for IPTV2.

In the evaluation, we compare the bigram models with some baseline methods from the current literature. These include the aforementioned DCMLDA model, and a version of the LDA where, for each user, the tokens represent (unordered) bigrams rather than single item occurrences. This is in practice a preprocessing of the data, which produces a different representation of the dataset upon which the standard LDA model is trained. Clearly, the ranking function has to be tuned accordingly.

We also provide two further baselines. The first one is a simple bigram model where the probability of occurrence of an item is modeled as $P(w_n) = \lambda f_{w_n} + (1 -$

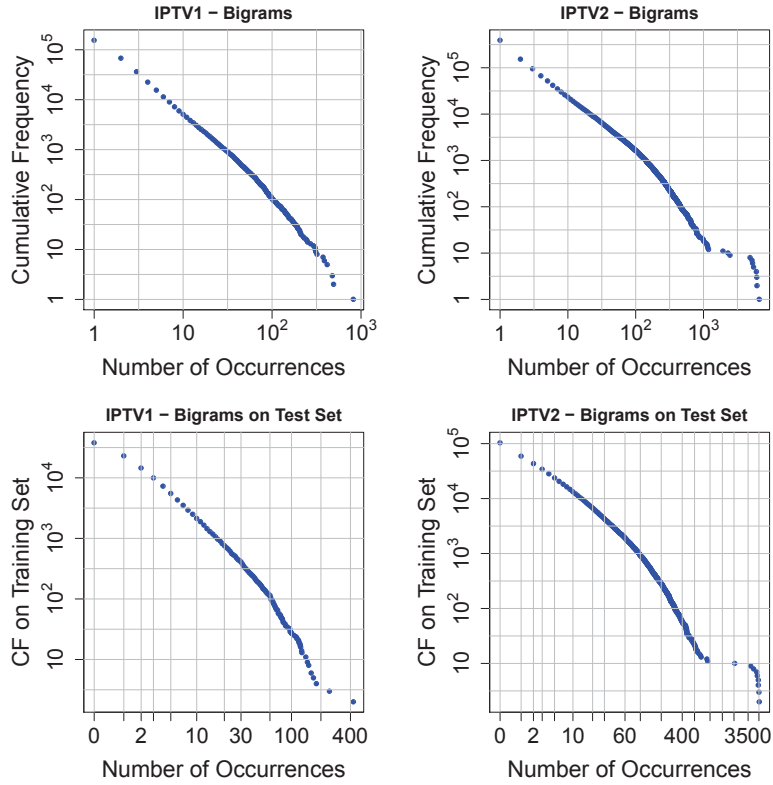


Fig. 4.4: Distributions of bigrams on real-life datasets

$\lambda)f_{w_n|w_{n-1}}$. Here, f_i is the relative frequency of i in the training set, whereas $f_{i|j}$ represents the same frequency conditioned to a preceding occurrence of j in the sequence. The λ parameter weighs the importance of the two components, and is tuned in a way proportional to the frequency of i , as typically low-frequency items do not provide a reliable estimates of the sequential part.

Finally, we also compare the proposed models to a baseline rooted on matrix factorization [61, 65]. The basic idea here is to exploit the matrix factorization for ranking, e.g., by providing an estimate of the probability of the item appearance [66]. There are some issues to consider when applying matrix factorization to the case at hand. In our context, matrix factorization is aimed at modeling item occurrence rather than an explicit rating. In this respect, non-occurrence of an item has a bi-valent interpretation, either as unknown (the user did not consider the item yet), or negative (she does not prefer it at all). Thus, the traditional approaches based on explicit preference (such as [79]) cannot be applied. We experimented with several specific techniques, including [57, 86] and the standard SVD model. In the following,

we report the results of the SVD⁶, that still outperforms all the other methods, as a confirmation of the findings in [5, 25].

Results

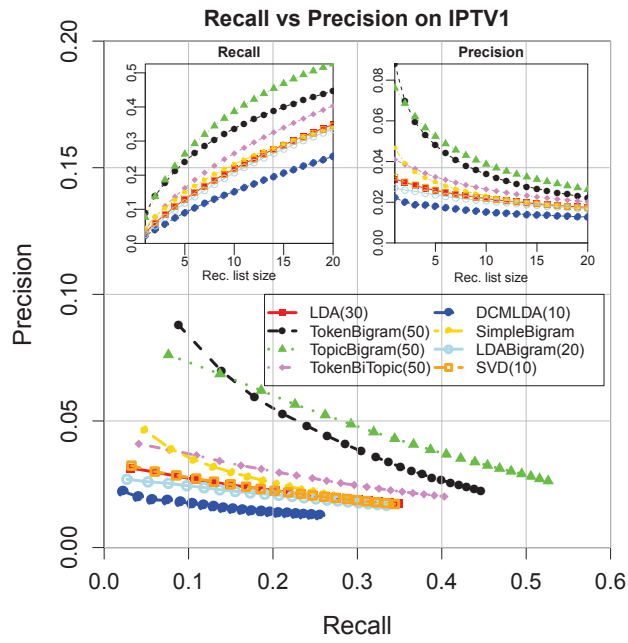
Fig. 4.5 summarizes the results in recommendation accuracy achieved over the two considered datasets. For each model, the optimal number of topics is given in brackets.

On both datasets, the proposed models improve the baselines. Concerning IPTV1, both TopicBigram and TokenBigram achieve a significant margin with respect to the other competitors. On IPTV2, TokenBigram outperforms TopicBigram, which is still the runner-up performer.

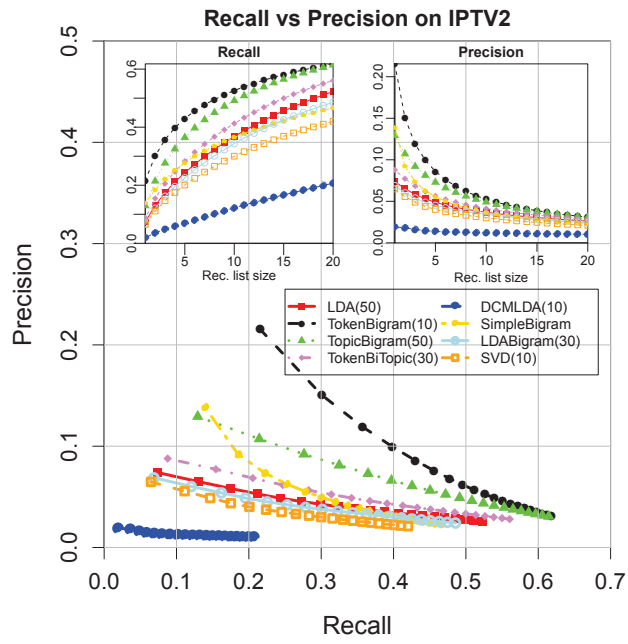
In summary, the results suggest that:

- The underlying assumption within TokenBitopic does not involve a remarkable increase of the predictive capabilities of the model. In practice, the topic structure of the TokenBitopic model can be “simulated” by an LDA model with a quadratic number of topics. As a result, the model seems more prone to overfitting;
- Contextual information, with particular reference to sequence modeling, provides a substantial contribution to recommendation accuracy. This is proven not only by the models proposed in this chapter: even the SimpleBigram baseline model achieves remarkable accuracy. In particular, when the recommendation list is relative small, the latter achieves an accuracy comparable to TokenBitopic. As a matter of fact, all the sequential approaches seem to provide a better estimate of the selection probability for the user’s next choice;
- There is a strict correlation between the frequencies exhibited by bigrams and the performance of the TokenBigram model. IPTV2 exhibits more frequent bigrams, and hence it is more likely to boost the performances of the TokenBigram model. By the converse, the TopicBigram exhibits a better capability in generalizing the dependency between the previous hidden context and the next choice. Geometrically, while the TokenBigram model focuses exclusively a restricted area of the topic space, induced by considering only the previous item, the TopicBigram model is actually able to identify larger homogeneous region within the topic space and to estimate the connections (transition probabilities) between them;
- Among the competitors, DCMLDA is rather weak. This is somehow surprising, considering that DCMLDA exhibits the best perplexity in the previous sets of experiments. A viable explanation of this dichotomy can be found in the nature of the sequential data explored here, which does not necessarily support burstiness: notably in a movie rental scenario, once a movie is rented by a user, it is unlikely that it is rented again in the future;
- LDABigram does not provide a substantial improvement either. Again, this is unexpected, in some sense, as bigrams can be considered contextual information

⁶ Based on the SVDLIBC implementation, <http://tedlab.mit.edu/~dr/SVDLIBC/>. The other matrix factorization methods were obtained from the Graphlab Library, <http://graphlab.org/>.



(a)



(b)

Fig. 4.5: Recommendation accuracy

as well. It seems that, when bigrams are introduced without an ordering relationship, the resulting ranking function is weakened.

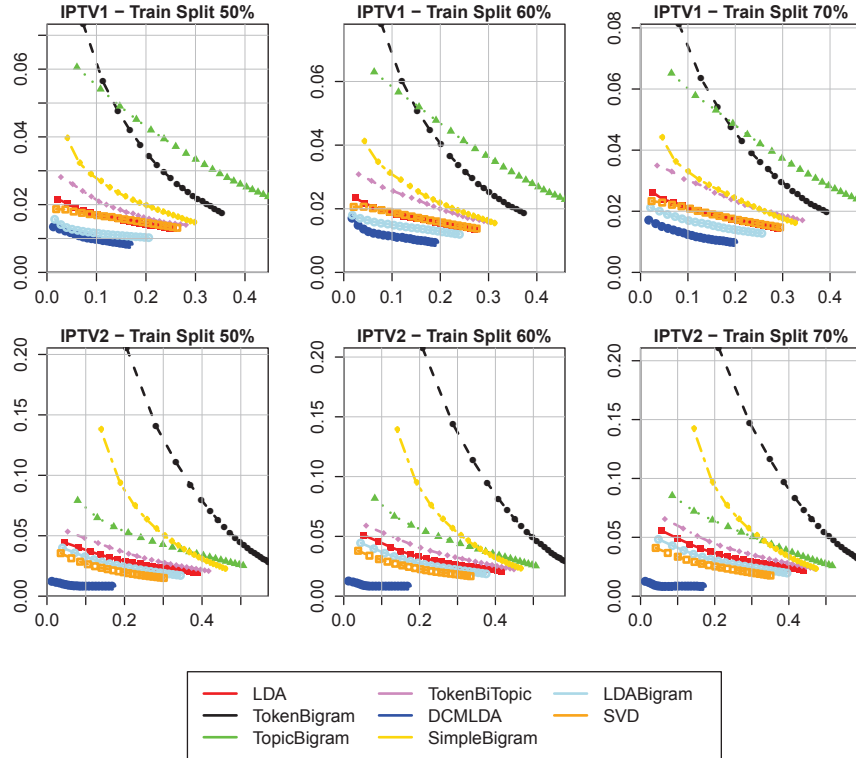


Fig. 4.6: Precision and recall for different training splits

In order to analyze the stability of the results, we perform some further experiments. First, we analyze the robustness of the previous experiment with regards to different training/test splits. Fig. 4.6 shows the precision/recall results on three further batches where each user sequence is split respectively to 50%, 60% and 70% of the size. In these plots, both TokenBigram and SimpleBigram tend to provide stable results, especially on IPTV2. All other methods seem to suffer the shrinking of the training partition.

In a second batch of experiments, we are interested in analyzing the robustness of the results with regards to random variations of the datasets. To this purpose, we repeat the above experiment on several random samples of the original dataset, where each sample includes 50% of the whole user set. Training and test sets for each sample are obtained by splitting each sequence with the standard 80-20 per-

centages. Fig. 4.7 shows average recall, as well as the intervals of variability. It is

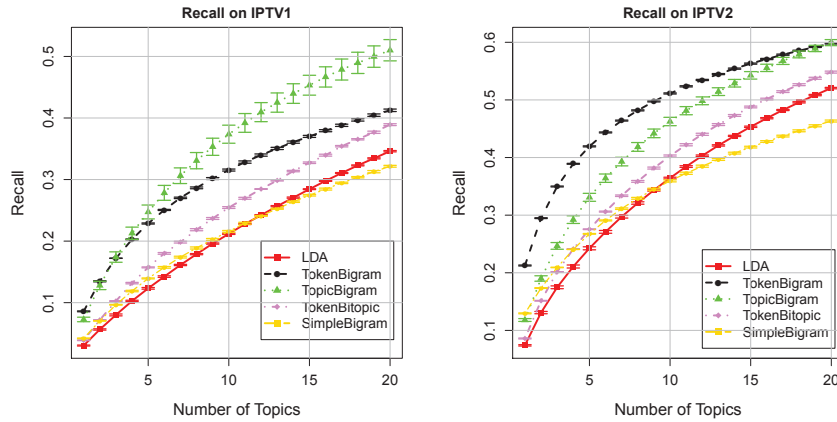


Fig. 4.7: Recall on random selections of users

worth noticing that the TopicBigram model exhibits the highest variations (especially on IPTV1). Notwithstanding, the performances of Fig. 4.5 are confirmed, thus witnessing a viable robustness of the proposed methods.

Finally, we confront in Fig. 4.8 the performance with regards to the number of latent factors, with a recommendation list fixed at size 20. TokenBitopic expresses a wide range of variability in IPTV1, and tends to improve with an increasing number of topics. The other models are stable, and in general do not show a large variance. On IPTV2, TopicBigram shows a progressive increase. However, the slope is progressively decreasing and hence we can expect a maximum on 50 topics. As for the competitors, SVD degrades as long as the number of latent factors is increased: a clear sign of overfitting (as also well-known from the literature). It is worth noticing that, albeit stabler, other matrix factorization approaches based on regularization (not reported here) are still weaker than SVD.

The results presented above experimentally show the effectiveness of sequential topic models in predicting future users' choices. However those models increase significantly the number of parameters to be learned and this implies an increase in the learning time. In Fig. 4.9 we plot the learning time (5000 Gibbs Sampling iterations) for different numbers of topics. Again, TopicBigram exhibits a quadratic behavior, due to the Markovian dependency among topic.

The last two plots in Fig. 4.9 highlight the contribution of asymmetric priors in the learning process. As expected, asymmetric priors significantly improve the accuracy. However, the learning time is greatly affected, as learning these parameters requires a further iterative fix point procedure to embed in the main algorithm, as explained in Section 4.4.2.

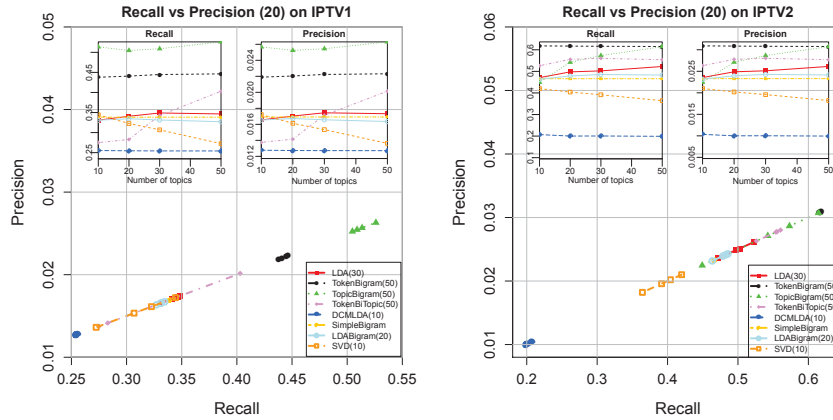


Fig. 4.8: Recall(20) and Precision(20) of the considered approaches varying the number of topics

4.7 Other sequential approaches

In this chapter we studied and extended Probabilistic topic models. These models provide a powerful and effective framework to discover the hidden thematic structure in large collection of documents [11]. As we aforesaid, the key idea is that each document/user may exhibit multiple topics/interest, where each topic is defined by a distribution over the dictionary which favors the occurrence of some semantic-related words over others. This *mixed membership assumption* can be verified, e.g., on real-world scenarios where each document focuses on a limited number of themes, some of them are dominant and others are subsidiary, and in the context of each theme some words are more relevant than other. Assuming that the mixed membership proportions and the topic distribution are known, the generative process for a document iterates the following procedure: (i) sample a topic according to the characterizing mixed proportion; (ii) randomly choose a word according to the topic distribution. The generative process is strongly based on a “bag-of-words” assumption. Even if this assumption may sound unrealistic, this modeling works really good in practice. The definition of the topic space and of the projection of each document into this space, provide an effective tool to infer the *semantic concept* of each document or, generally, entity. These approaches support three main tasks [42]: topic extraction, word sense disambiguation and prediction.

- *Topic Extraction*: given a document, infer its topic distribution and its dominant topics;
- *Word sense disambiguation*: each topic is defined as a distribution of words which tend to occur in the same document. Thus, the statistical analysis of co-occurrence, collects into the same topic words that refer to the same semantic concept;

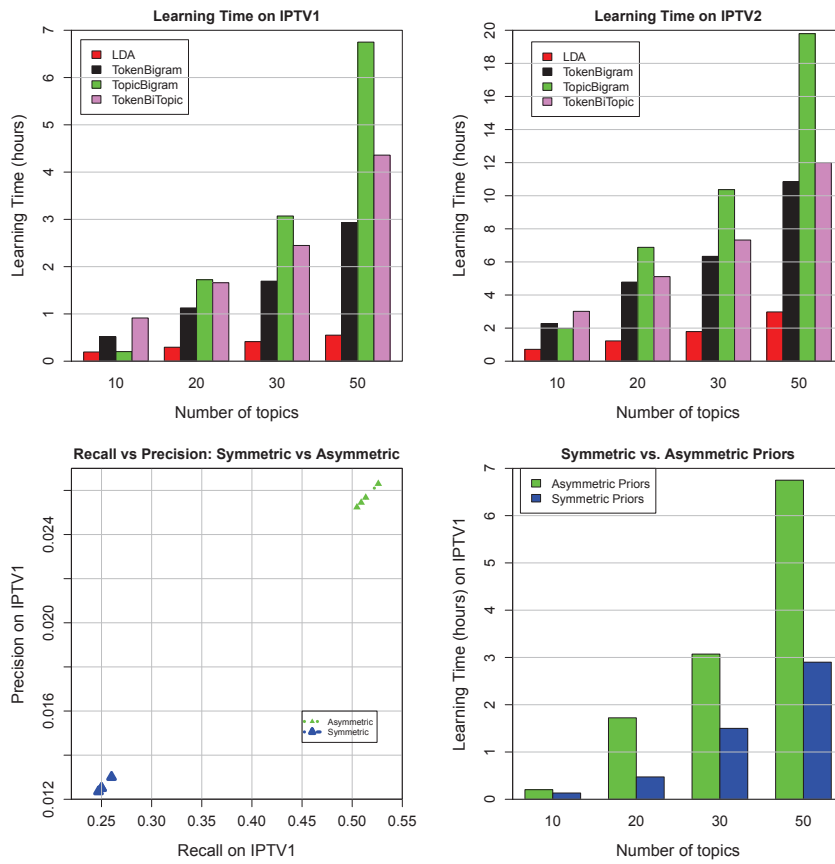


Fig. 4.9: Learning time of the models on IPTV1 and IPTV2 (first row); influence of the hyper parameters (second row)

- *Prediction*: given the current word and the underlying topic, we can predict which words are more likely to follow in the considered document. This is the task in which we focused in this chapter.

Among all the different contexts in which these approaches have achieved significant results, in this chapter we considered the application of probabilistic topic models to the recommendation problem [54]. In this setting we exploited the prediction capabilities of topic models, since we are asked to predict which items the user will select next, given her past preferences. As explained in previous sections, this choice is motivated by some interesting recent findings [5] which can be summarized as follows: (i) the item-selection probability computed for each user is a key component for generating accurate item-ranking functions; (ii) among all competitors, LDA provides the best results measured in precision and recall of the recommenda-

tion list. These promising results motivated us in exploring extensions of topic models (see Section 4.4) and we shown, after an experimental evaluation over real-life datasets, that they provide better representation of the inherent sequential correlation between items, and thus provide better performances in predictions (see Section 4.6). In this section we are going to briefly review related state-of-the-art probabilistic approaches to sequence data modeling, mainly focusing on topic approaches.

A simple approach to model sequential data within a probabilistic framework has been proposed in [19]. In this work, authors present a framework based on mixtures of Markov models for clustering and modeling of web site navigation logs, which is applied for clustering and visualizing user behavior on a web site. Although simple, the proposed model suffers of the limitation that a single latent topic underlies all the observation in a single sequence. This approach has been overtaken by other methods based on latent semantic indexing and LDA. In [99, 102], for example, the authors propose extension of the LDA model which assume a first-order Markov chain for the word generation process. In the resulting *Bigram Model (BM)* (see Section 4.4) and *Topical n-grams*, the current word depends on the current topic and the previous word observed in the sequence.

The N -gram modeling can be extended by considering different kind of dependencies between the hidden states of the model. These kind of dependencies are formalized by exploiting *Hidden Markov models* [10], which are a general reference framework both for modeling sequence data and for natural language processing [63]. HMMs assume that sequential data are generated using a Markov chain of latent variables, with each observation conditioned on the state of the corresponding latent variable. The resulting likelihood can be interpreted as an extension of a mixture model in which the choice of mixture components for each observation is not selected independently but depends on the choice of components for the previous observation. In [43], authors delve in this direction, and propose an *Hidden Topic Markov Model (HTMM)* for text documents. HTMM defines a Markov chain over latent topics of the document. The corresponding generative process, depicted in Fig. 4.10(a), assumes that all words in the same sentence share the same topic, while successive sentences can either rely on the previous topic, or introduce a new one. The topics in a document form a Markov chain with a transition probability that depends on a binary topic transition variable ψ . When $\psi = 1$, a new topic is drawn for the n -th sentence, otherwise the same previous topic is used.

The *LDA Collocation Model* [42] introduces a new set of random variables (for bigram status) x which denotes whether a bigram can be formed with the previous word token. More specifically, as represented in Fig. 4.10(b), the generative process specifies for each word both a topic and a collocation status.

- if $x_i = 1$ then w_i is part of a collocation and thus is generated by sampling from a distribution conditioned on the previous word $P(w_i|w_{i-1}, x_i = 1)$;
- otherwise, w_i is sampled from a distribution associated to the current topic $P(w_i|z_i, x_i = 0)$

The collocation status adds a more realistic than Wallach's model which always generates bigrams and, according to this formulation, the distribution on bigram does not depend on the topic. The introduction of the collocation status enrich the generative semantic of the model and this idea can be applied to all the approaches proposed in Section 4.4.

All the previously discussed models approach the problem of sequence modeling by inferring the underlying latent topic and then generate a sequence of words according to this distribution. This perspective, does not take into account the fact that words in a text document may exhibit both syntactical and semantic correlations. A *Composite Model*, which captures both semantic and syntactic roles, has been proposed in [41]. The graphical model for the generation of a document, given in Fig. 4.10(c), clarify this concept. The semantic/syntactic dependencies among words are modeled by employing two different latent variables, namely Z and C ; while the semantic layer follows a simple LDA model, the syntactic one is instantiated by modeling transitions between the set of classes C through an hidden Markov model. One of these classes correspond to the semantic class and, when is observed, enable the generation of the word according the current topic. Other classes capture word co-occurrences that are due to syntactic aspects of the modeled language.

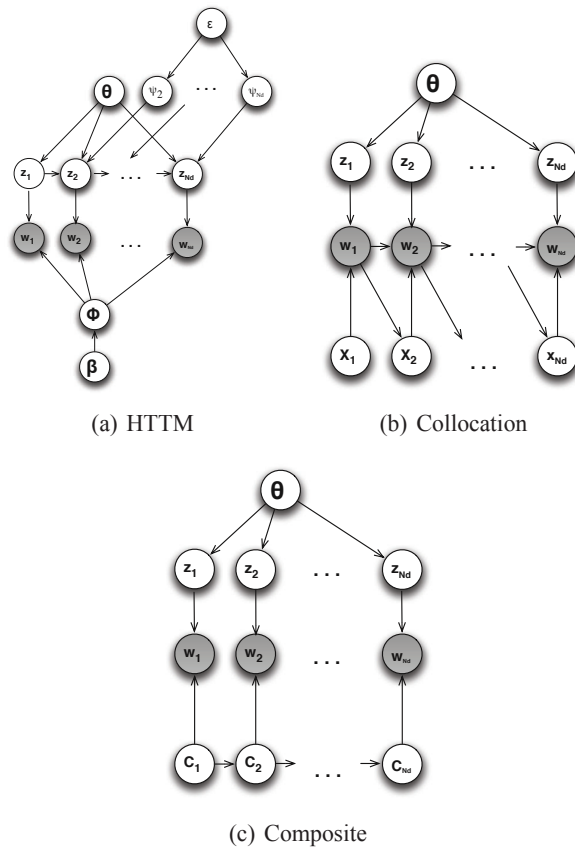


Fig. 4.10: HTMM, Collocation and Composite graphical model for the generation of a document

Conclusion

In this thesis we have presented an integrated and innovative framework for the development of web applications with advanced features. The whole framework is composed by a powerful set of tools and offers a wide range of features: Document and Content Management, Social Cooperation, Knowledge Discovery, Business Process Management.

The core of the framework, *Borè*, allows Web information to be defined, organized, stored, queried and displayed as customizable resources and relations, and it directly supports social networks, which spontaneously arise when users share resources among each others.

An extension of the *Borè* platform, *Caldera*, introduces innovative features with the aim of defining a complete and unified framework. *Caldera* is a platform for managing and analyzing collaborative process, which integrates an advanced recommender system. *Caldera* is innovative under different perspectives, since it allows (i) the definition and the management of semi-structured contents and (ii) to realize Social Networks and Social Cooperations; (iii) it is integrated with both recommender systems and (iv) with innovative features of process mining through log analysis. The last two topics were widely investigated and innovative solutions have been introduced. A thorough experimental evaluation over real-world data shows the performance advantages of the proposed approaches. The *Caldera* architecture is reported in Fig. 5.1.

We have widely discussed the advanced business process management techniques exploited by *Caldera*; in particular we have focused on Process Discovery and Performance Prediction issues. Concerning Process Discovery we have presented a survey of the state-of-the-art techniques that aim at automatically extracting behavioral process models from log data. Regarding Performance Prediction, after reporting leading techniques, we have sketched the main ideas and contributions of an innovative approach presented in [8, 9] to discover predictive models for run-time support. In these works, we have combined performance prediction with a predictive clustering technique; this brings substantial gain in terms of readability and accuracy. Moreover the proposed technique frees the analyst from the burden of explicitly set-

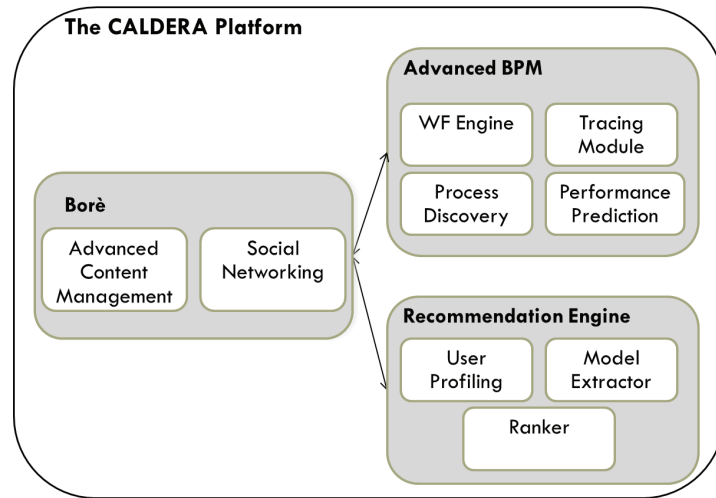


Fig. 5.1: Caldera architecture

ting the abstraction level which is determined, instead, in a data-driven way. Work presented in [8] has been awarded with the *Best Paper Award* during the 15th International Conference on Enterprise Information Systems (ICEIS13).

Concerning Recommendation and Knowledge Discovery, we have focused on probabilistic approaches to recommendation and we have presented our extensions to classic probabilistic topic models, introducing causality and dependency through a sequential approach. We have studied three extensions of the LDA model which relax the bag-of-words assumption by hypothesizing that the current observation depends on previous information. For each of the proposed models we have provided a Gibbs Sampling parameter estimation procedure and an experimental evaluation was accomplished by studying the models both from a model fitting and an applicative perspective. We have found that the proposed models provide a better framework for modeling contextual information in a recommendation scenario, when the data exhibit intrinsic temporal dependency.

In short, our contributions in this direction can be summarized as follows:

1. we have proposed a unified probabilistic framework to model dependency in preference data, and instantiate the framework in accordance to different assumptions on the sequentiality of the underlying generative process;
2. we have studied and experimentally compared the proposed models, highlighting relative advantages and weaknesses;
3. we have studied how to adapt this framework to support a recommendation scenario. In particular, for each of the proposed models, we have provided the rela-

tive ranking function that can be used to generate personalized and context-aware recommendation lists;

4. we have shown that the proposed sequential modeling of preference data better models the underlying data, as it allows more accurate recommendations in terms of precision and recall.

Besides the scenarios investigated, we believe that this topic is worth extending in two main directions. On the one side, it would be interesting to generalize the notion of “contextual information”: in our approach, a context was represented by temporal dependency. However, there are other observable features that can contribute in the likelihood of observing an item in a user’s trace, such as geographical location, tags, etc. Even further, interaction of a user in a social network is having an increasing impact in user’s behavior. Analyzing the influence of the neighbors in a network can help better evaluate both the temporal dependencies and the likelihood of an item to be selected.

References

1. Agarwal D., Chen B.: Flda: matrix factorization through latent dirichlet allocation. Proc. of the 3rd ACM Int. Conf. on Web Search and Data Mining (WSDM10), pp. 91–100, 2010.
2. Agrawal R., Imielinski T., Swami A.: Mining association rules between sets of items in large databases. Proc. of the 1993 ACM SIGMOD Int. Conf. on Management of Data (SIGMOD93), pp. 207–216, 1993.
3. Agrawal R., Srikant R.: Fast algorithms for mining association rules. Proc. of the 20th Int. Conf. on Very Large Databases (VLDB94), pp. 487–499, 1994.
4. Bambini R., Cremonesi P., Turrin R.: A recommender system for an iptv service provider: a real large-scale production environment. Recommender Systems Handbook, pp. 299–331. Springer, 2011.
5. Barbieri N., Manco G.: An analysis of probabilistic methods for top-n recommendation in collaborative filtering. Proc. of the 2011 European Conf. on Machine learning and knowledge discovery in databases (ECML-PKDD11), pp. 172–187, 2011.
6. Barbieri N., Manco G., Ortale R., Ritacco E.: Balancing prediction and recommendation accuracy: Hierarchical latent factors for preference data. Proc. of the 2012 SIAM Int. Conf. on Data Mining (SDM12), 2012.
7. Bennett J., Lanning S., Netflix N.: The netflix prize. KDD Cup and Workshop in conjunction with KDD, 2007.
8. Bevacqua A., Carnuccio M. Folino F., Guarascio M., Pontieri L.: A data-adaptive trace abstraction approach to the prediction of business process performances. Proc. of the 15th Int. Conf. on Enterprise Information Systems (ICEIS13), 2013.
9. Bevacqua A., Carnuccio M. Folino F., Guarascio M., Pontieri L.: Adaptive trace abstraction approach for predicting business process performances. Proc. of the 21st Italian Symposium on Advanced Database Systems (SEBD13), 2013.
10. Bishop C.: Pattern recognition and machine learning. Springer, 2006.
11. Blei D.M.: Introduction to probabilistic topic models. Communications of the ACM, 2011.
12. Blei D.M., McAuliffe J.D.: Supervised topic models. Proc. of the 22nd Neural Information Processing Systems (NIPS08), 2008.
13. Blei D.M., Ng A.Y., Jordan M.I.: Latent dirichlet allocation. The Journal of Machine Learning Research, vol. 3, pp. 993–1022, 2003.
14. Blockeel H., De Raedt L.: Top-down induction of first-order logical decision trees. Artificial Intelligence 101 (1–2), pp. 285–297, 1998.

15. Blockeel H., De Raedt L., Ramon J.: Top-down induction of clustering trees. Proc. of the 15th Int. Conf. on Machine Learning (ICML98), pp. 55–63, 1998.
16. Bondy A., Murty U.S.R.: Graph theory. 3rd Corrected Printing, Springer, 2008.
17. Breese J.S., Heckerman D., Kadie C.: Empirical analysis of predictive algorithms for collaborative filtering. Proc. of the 14th Conf. on Uncertainty in Artificial Intelligence (UAI98), pp. 43–52, 1998.
18. Breiman L., Friedman J., Olshen R., Stone C.: Classification and regression trees. Wadsworth and Brooks, Monterey, CA, 1984.
19. Cadez I., Heckerman D., Meek C., Smyth P., White S.: Visualization of navigation patterns on a web site using model-based clustering. Proc. of the 6th ACM SIGKDD Int. Conf. on Knowledge discovery and data mining (KDD00), pp. 280–284, 2000.
20. Ceri S., Fraternali P., Bongio A., Brambilla M., Comai S., Matera M.: Designing Data-Intensive Web Applications. Morgan-Kaufmann, 2002.
21. Cesario E., Folino F., Ortale R.: Putting enhanced hypermedia personalization into practice via web mining. Proc. of the 15th Int. Conf. on Database and Expert Systems Applications (DEXA04), pp. 947–956, 2004.
22. Chang F., Dean J., Ghemawat S., Hsieh W. C., Wallach D. A., Burrows M., Chandra T., Fikes A., Gruber R. E.: Bigtable: a distributed storage system for structured data. Proc. of the 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI06), pp. 205–218, 2006.
23. Chen C.W.K., Yun D.Y.Y.: Discovering process models from execution history by graph matching. Proc. of the 4th Int. Conf. on Intelligent Data Engineering and Automated Learning (IDEAL03), pp. 887–892, 2003.
24. Conforti R., Fortino G., La Rosa M., ter Hofstede A.H.M.: History-aware, real-time risk detection in business processes. Proc. of the 19th Int. Conf. on Cooperative Information Systems (CoopIS11), pp. 100–118, 2011.
25. Cremonesi P., Koren Y., Turrin R.: Performance of recommender algorithms on top-n recommendation tasks. ACM RecSys, pp. 39–46, 2010.
26. Cremonesi P., Turrin R.: Analysis of cold-start recommendations in iptv systems. Proc. of the 3rd ACM Conf. on Recommender systems (RecSys09), pp. 233–236, 2009.
27. Das A., Datar M., Garg A., Rajaram S.: Google news personalization: scalable online collaborative filtering. Proc. of the 16th Int. Conf. on World Wide Web (WWW07), pp. 271–280, 2007.
28. de la Vara J.L., Ali R., Dalpiaz F., Sanchez J., Giorgini P.: COMPRO: a methodological approach for business process contextualisation. Proc. of the 18th Int. Conf. on Cooperative Information Systems (CoopIS10), pp. 132–149, 2010.
29. de Medeiros A.K.A. et al.: Process mining based on clustering: a quest for precision. In Business Process Management Workshops (BPM07), pp. 17–29, 2007.
30. de Medeiros A.K.A., van Dongen B.F., van der Aalst W.M.P., Weijters A.J.M.M.: Process mining: extending the α -algorithm to mine short loops. Technical Report WP 113. Eindhoven University of Technology, 2004.
31. de Medeiros A.K.A., Weijters A.J.M.M., van der Aalst W.M.P.: Genetic process mining: an experimental evaluation. Data Mining and Knowledge Discovery, vol. 14(2), pp. 245–304, 2007.
32. Deerwester S.: Improving information retrieval with latent semantic indexing. Proc. of the 51st ASIS Annual Meeting (ASIS88), vol. 25, 1988.
33. DLAI Group: CLUS: a predictive clustering system. Available at <http://dtai.cs.kuleuven.be/clus/>, 1998.
34. Doyle G., Elkan C.: Accounting for burstiness in topic models. Proc. of the 26th Int. Conf. on Machine Learning (ICML09), p. 281–288, 2009.

35. Draper N.R., Smith H.: Applied regression analysis. Wiley Series in Probability and Statistics, 1998.
36. Folino F., Guarascio M., Pontieri L.: Discovering context-aware models for predicting business process performances, Proc. of the 20th Int. Conf. on Cooperative Information Systems (CoopIS12), 2012.
37. Georgakopoulos D., Hornick M., Sheth A.: An overview of workflow management: from process modeling to workflow automation infrastructure. Distributed and Parallel Databases, vol. 3(2), pp. 119–153, 1995.
38. Ghionna L., Greco G., Guzzo A., Pontieri L.: Outlier detection techniques for process mining applications. Proc. of the 17th Int. Symposium on Foundations of Intelligent Systems (ISMIS08), pp. 150–159, 2008.
39. Goldberg D., Nichols D., Oki B., Terry D.: Using collaborative filtering to weave an information tapestry. Communications of the ACM, vol. 35, pp. 61–70, 1992.
40. Greco G., Guzzo A., Pontieri L., Sacc D.: Discovering expressive process models by clustering log traces. IEEE Trans. on Knowledge and Data Engineering, vol. 18(8), pp. 1010–1027, 2006.
41. Griffiths T., Steyvers M., Blei D., Tenenbaum J.B.: Integrating topics and syntax. In Advances in Neural Information Processing Systems, vol. 17, pp. 537–544, 2005.
42. Griffiths T.L., Steyvers M., Tenenbaum J.B.: Topics in semantic representation. Psychological Review 114, 2007.
43. Gruber A., Weiss Y., Rosen-Zvi M.: Hidden topic markov models. Journal of Machine Learning Research, vol. 2, pp. 162–170, 2007.
44. Günther C.W., van der Aalst W.P.M.: Fuzzy mining: adaptive process simplification based on multi-perspective metrics. Proc. of the 5th Int. Conf. on Business Process Management (BPM07), pp. 328–343, 2007.
45. Hamilton J. : Perspectives: one size does not fit all. Google Retrieved, November 13, 2009.
46. Han J., Pei J., Yi Y.: Mining frequent patterns without candidate generation. Proc. Int. ACM Conf. on Management of Data (SIGMOD00), pp. 1–12, 2000.
47. Harlde W.: Applied nonparametric regression. Cambridge University Press, 1990.
48. Hardle W., Mammen E.: Comparing nonparametric Versus parametric regression fits. The Annals of Statistics 21, pp. 1926–1947, 1993.
49. Heinrich G.: Parameter estimation for text analysis. Technical Report, University of Leipzig, 2008.
50. Herbst J., Karagiannis D.: Integrating machine learning and workflow management to support acquisition and adaptation of workflow models. Journal of Intelligent Systems in Accounting, Finance and Management, vol. 9, pp. 67–92. 2000.
51. Herbst J., Karagiannis D.: Workflow mining with InWoLvE. Computers in Industry. Special Issue: Process/Workflow Mining, vol. 53(3), pp. 245–264, 2003.
52. Herlocker J., Konstan J.A., Riedl J.: An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. Information Retrieval, vol. 5, pp. 287–310, 2002.
53. Hofmann T.: Collaborative filtering via gaussian probabilistic latent semantic analysis. Proc. of the ACM SIGIR Conf. on Research and development in informaion retrieval (SIGIR03), pp. 259–266, 2003.
54. Hofmann T.: Latent semantic models for collaborative filtering. ACM Trans. on Information Systems vol. 22(1), pp. 89–115, 2004.
55. Hofmann T.: Learning what people (don't) want. Proc. of the 12th European Conf. on Machine Learning (EMCL01), pp. 214–225, 2001.

56. Hofmann T., Puzicha J.: Latent class models for collaborative filtering. Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI99), pp. 688–693, 1999.
57. Hu Y., Koren Y., Volinsky C.: Collaborative filtering for implicit feedback datasets. Proc. of the 8th IEEE Int. Conf. on Data Mining (ICDM08), pp. 263–272, 2008.
58. Jagadeesh Chandra Bose R.P., van der Aalst W.P.M.: Context aware trace clustering: towards improving process mining results. Proc. of the 2009 SIAM Int. Conf. on Data Mining (SDM09), pp. 401–412, 2009.
59. Jagadeesh Chandra Bose R.P., van der Aalst W.P.M.: Trace clustering based on conserved patterns towards achieving better process models. Proc. of the 5th Int. Workshop on Business Process Intelligence (BPI09), 2009.
60. Junginger S., Kuhn H., Strobl R., Karagiannis D.: Ein geschäftsprozessmanagementwerkzeug der nächsten generation — ADONIS: Konzeption und anwendungen. *Wirtschaftsinformatik*, vol. 42(3), pp. 392–401, 2000.
61. Koren Y., Bell R., Volinsky C.: Matrix factorization techniques for recommender systems. *IEEE Computer*, vol. 42(8), pp. 30–37, 2009.
62. Lieberman H.: Letizia: an agent that assists web browsing. Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI95), pp. 924–929, 1995.
63. Manning C.D., Schütze H.: Foundations of statistical natural language processing. MIT Press, Cambridge, MA, USA, 1999.
64. Marlin B.: Collaborative filtering: a machine learning perspective. Technical report, Department of Computer Science University of Toronto, 2004.
65. Menon A., Elkan C.: Link prediction via matrix factorization. Proc. of the 2011 European conference on Machine learning and knowledge discovery in databases (ECML-PKDD11), pp. 437–452, 2011.
66. Menon A., Elkan C.: Predicting labels for dyadic data. *Data Mining and Knowledge Discovery*, vol. 21(2), pp. 327–343, 2010.
67. Minka T.P.: Estimating a Dirichlet distribution. Technical Report, Microsoft Research, 2000.
68. Mitchell J.: 10 concepts in object-oriented languages. Concepts in programming language. Cambridge University Press, p. 287, 2002.
69. Nakatumba J., van der Aalst W.M.P.: Analyzing resource behavior using process mining. In *Business Process Management Workshops (BPM09)*, pp. 69–80, 2009.
70. Pazzani M.J., Billsus D.: Content-based recommendation systems. The Adaptive Web, vol. 4321 of Lecture Notes in Computer Science, pp. 325–341, 2007.
71. Peters I., Becker P.: Folksonomies. Indexing and retrieval in Web2.0. *Knowledge & Information: Studies in Information Science*, 2009.
72. Pink Daniel H.: Folksonomy. *New York Times*, December 11, 2005.
73. Polyvyanyy A., Smirnov V., Weske M.: The triconnected abstraction of process models. Proc. of the 7th Int. Conf. on Business Process Management (BPM09), pp. 229–244, 2009.
74. Quinlan J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., 1993.
75. Quinlan R.J.: Learning with continuous classes. Proc. of the 5th Australian Joint Conf. on Artificial Intelligence (AI92), pp. 343–348, 1992.
76. Racine J., Li Q.: Nonparametric estimation of regression functions with both categorical and continuous data. *Journal of Econometrics*, vol. 119(1), pp. 99–130, 2004.
77. Ricci F., Rokach L., Shapira B., Kantor P.B (editors): *Recommender Systems Handbook*, Springer, 2011.
78. Ristad E.S., Yianilos P.N.: Learning string-edit distance. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20(5), pp. 522–532, 1998.

79. Salakhutdinov R., Mnih A.: Probabilistic matrix factorization. Proc. of the 21st Conf. on Neural Information Processing Systems (NIPS07), 2007.
80. Sarwar B.M., Karypis G., Konstan J.A., Reidl J.: Item-based collaborative filtering recommendation algorithms. World Wide Web, pp. 285–295, 2001.
81. Schafer J.B., Konstan J.A., Riedl J.: E-commerce recommendation applications. Data Mining and Knowledge Discovery, vol. 5(1–2), pp. 115–153, 2001.
82. Schimm G.: Mining most specific workflow models from event-based data. Proc. of the 1st Int. Conf. on Business Process Management (BPM03), pp. 25–40, 2003.
83. Schonenberg H., Weber B., van Dongen B.F., van der Aalst W.M.P.: Supporting flexible processes through recommendations based on history. Proc. of the 6th Int. Conf. on Business Process Management (BPM08), pp. 51–66, 2008.
84. Schuldts H., Alonso G., Beerl C., Schek H.: Atomicity and isolation for transactional processes. ACM Trans. on Database Systems, vol. 7(1), pp. 63–116, 2002.
85. Scott M. L.: Programming language pragmatics. Edition 2, Morgan Kaufmann, p. 470, 2006.
86. Sindhwani V., Bucak S., Hu J., Mojsilovic A.: One-class matrix completion with low-density factorizations. Proc. of the 10th IEEE Int. Conf. on Data Mining (ICDM10), pp. 1055–1060, 2010.
87. Song M., Gnther C.W., van der Aalst W.P.M.: Trace clustering in process mining. In Business Process Management Workshops (BPM08), pp. 109–120, 2008.
88. Stern D.H., Herbrich R., Graepel T.: Matchbox: large scale online bayesian recommendations. Proc. of the 18th Int. Conf. on World Wide Web (WWW09), pp. 111–120, 2009.
89. van der Aalst W.M.P.: The application of Petri nets to workflow management. Journal of Circuits, Systems, and Computers, vol. 8(1), pp. 21–66, 1998.
90. van der Aalst W.M.P. *et al.*: ProM 4.0: comprehensive support for real process analysis: Proc. of the 28th Int. Conf. on Applications and Theory of Petri Nets and Other Models of Concurrency (ICATPN07), pp. 484–494, 2007.
91. van der Aalst W.M.P., Schonenberg M.H., Song M.: Time prediction based on process mining. Information Systems, vol. 36(2), pp. 450–475, 2011.
92. van der Aalst W.M.P., van Dongen B.F., Herbst J., Maruster L., Schimm G., Weijters A.J.M.M.: Workflow mining: a survey of issues and approaches. Data & Knowledge Engineering, vol. 47(2), pp. 237–267, 2003.
93. van der Aalst W.M.P., Weijters A.J.M.M., Maruster L.: Workflow mining: discovering process models from event logs. IEEE Trans. on Knowledge and Data Engineering, vol. 16(9), pp. 1128–1142, 2004.
94. van Dongen B.F., Alves de Medeiros A.K., Verbeek H.M.W., Weijters A.J.M.M., van der Aalst W.M.P.: The ProM framework: a new era in process mining tool support. Proc. of the 26th Int. Conf. on Applications and Theory of Petri Nets (ICATPN05), pp. 444–454, 2005.
95. van Dongen B.F., Crooy R.A., van der Aalst W.M.P.: Cycle time prediction: when will this case finally be finished? Proc. of the 16th Int. Conf. on Cooperative Information Systems (CoopIS08), pp. 319–336, 2008.
96. van Dongen B.F., van der Aalst W.M.P.: A meta model for process mining data. Proc. of the CAiSE 2005 Workshops, pp. 309–320, 2005.
97. van Dongen B.F., van der Aalst W.M.P.: Multi-phase process mining: Aggregating instance graphs into EPCs and Petri Nets. In Proc. of the Int. Workshop on Applications of Petri Nets to Coordination, Workflow and Business Process Management (PNCWB05), 2005.
98. Verbeek H.M.W., Buijs J., van Dongen B. F., van der Aalst W.M.P.: XES, XESame, and ProM 6. Information Systems Evolution: CAiSE Forum 2010, vol. 72, pp. 60–75, 2010.

99. Wallach H.M.: Topic modeling: beyond bag-of-words. Proc. of the 23rd Int. Conf. on Machine learning (ICML06), pp. 977–984, 2006.
100. Wallach H., Mimno D., McCallum A.: Rethinking lda: why priors matter. Advances in Neural Information Processing Systems, vol. 22, pp. 1973–1981, 2009.
101. Wallach H., Murray I., Salakhutdinov R., Mimno D.: Evaluation methods for topic models. Proc. of the 26th Int. Conf. on Machine learning (ICML09), 2009.
102. Wang X.A.M., Wei X.: Topical n-grams: phrase and topic discovery, with an application to information retrieval. Proc. of the 7th IEEE Int. Conf. on Data Mining (ICDM07), pp. 697–702, 2007.
103. Weijters A.J.M.M., van der Aalst W.M.P.: Rediscovering workflow models from event-based data using little thumb. Integrated Computer-Aided Engineering, vol. 10(2), pp. 151–162, 2003.
104. Wen L., Wang J., Sun J.G.: Detecting implicit dependencies between tasks from event logs. Proc. of the 8th Asia-Pacific Web Conf (APWeb06), pp. 591–603, 2006.
105. Witten I.H., Frank E.: Data mining: practical machine learning tools and techniques. Second Edition, Morgan Kaufmann Publishers, 2005.
106. Zitnick C.L., Kanade T.: Maximum entropy for collaborative filtering. Proc. of the 20th Conf. on Uncertainty in Artificial Intelligence (UAI04), pp. 636–643, 2004.