UWS Academic Portal

TaxoPublish

Angermann, Heiko ; Ramzan, Naeem

Published in:
Expert Systems with Applications

DOI:
10.1016/j.eswa.2016.08.058

Published: 30/12/2016

Document Version
Peer reviewed version

TaxoPublish: Towards a solution to automatically personalize taxonomies in e-catalogs

Heiko Angermann[a,*], Naeem Ramzan[a]

[a] *University of the West of Scotland, School of Engineering and Computing, High Street, Paisley PA1 2BE, United Kingdom*

Abstract

Taxonomies are utilized in e-catalogs to facilitate customers navigating through a marketplace with the help of hierarchically structured concepts. However, when entering the e-catalog, each customer is shown the identical taxonomy regardless their individual requirements. Customers are distracted when navigating to preferred concepts as those are siblings of not required concepts. Provided progress in dynamic taxonomies, catalog segmentation, and personalized directories lacks in a fully automatic support for modifying the taxonomy according to the user's requirements. The existing works need an explicit user-query, are missing information about the domain, or require the modification through the provider. In this paper, *TaxoPublish* expert system based on logic programming is presented. The proposed system predicts the customers requirements for automatically modifying the taxonomy in B2B context. With *TaxoPublish*, retailers can now provide personalization in the form of personalized e-catalogs without any human effort, and without missing any information about the domain. *TaxoPublish* is using knowledge provided through a Customer Relationship Management system for predicting customers preferences, and knowledge of a Product Information Management system for performing taxonomic operations based on two novel types of taxonomic concepts. Through the usage of logic programming and the cross-platform database model, *TaxoPublish* can be applied as expert system over distributed and heterogeneous data warehouse architectures across various domains. The comprehensive experiments on two public and one private database show that *TaxoPublish* expert system is capable of fully-automatic taxonomy modification with an accuracy similar to the expert manual modifications.

*Corresponding author: Heiko Angermann, University of the West of Scotland,
School of Engineering and Computing, High Street, Paisley PA1 2BE, United Kingdom, Tel +44 (0) 141 848 3648
Email addresses: *B00274523@studentmail.uws.ac.uk (Heiko Angermann), naeem.ramzan@uws.ac.uk (Naeem Ramzan)*

Keywords: Taxonomy Overload, Electronic Catalog, Logic Programming, Personalized Directory, Electronic Commerce, Product Information Management

1. Introduction

Nowadays retailing markets are more diverse and fragmented than ever before, presenting early and potential customers with an overload of information (Liao et al. (2008); Álvaro Tejeda-Lorente et al. (2015)). Leading e-commerce applications (e.g. PrestaShop[1], or Magento[2]) provide two metadata techniques to help customers finding goods in e-catalogs when entering the retailing market. The lightweight method named folksonomy is applied to products, customer reviews, or images. It utilizes informal keywords created through the provider or customer for generating interlinked networks. As there are no restrictions for creating keywords, folksonomies contain semantic ambiguities and synonyms (Liang et al. (2010)). On the other hand, taxonomies, also called directories, are applied to model a field of interest in a formal way (Guarino et al. (2009)). This hierarchical representation of a domain has its merits for navigation, and for exploring similar products. However, as the creation of a taxonomy is restricted to the expert, or by referring to a standard taxonomy, every customer is treated with the same taxonomy when entering the digital marketplace. The semantic context weight (Viswanathan and Krishnamurthi (2012)) is too limited for highly required concepts, but too strong for low required concepts. This leads to a new challenge named taxonomy overload.

Besides the established techniques of catalog segmentation, where the customers are assigned to pre-defined sub catalogs (e.g. in Amiri (2006); Döring et al. (2006); George et al. (2013); Mahdavi et al. (2011); Xu et al. (2009, 2014)), and the works of dynamic taxonomies, which prune the taxonomy in response to a provided keyword (e.g. in Basu Roy et al. (2008); Bonino et al. (2009); Calegari and Pasi (2013); Gollub et al. (2014); Kumaraguru et al. (2014); Sacco et al. (2012); Tvarozek and Bielikova (2007); Vandic et al. (2012)), only a few research considers personalized directories, aiming to modify the taxonomy according to customers' requirements. Until now, this techniques have not affected real-world e-commerce applications because of two main reasons. Firstly, because of the modification has to be performed manually, which is inefficient for high-traffic retailing markets. Secondly, because of each modification is changing the semantics inside the taxonomy. For example, a not correct change of a super concept when moving sub concepts to a higher level, would hamper the customers in finding the desired products, as the initial label of

[1]https://www.prestashop.com/
[2]http://www.magento.com/

the super concept becomes inadequate.

For example, the logic programming and most comprehensive approach presented in Joh and Lee (2003) aims in overcoming the problems of dynamically changing e-catalogs in B2B. Their system provides different modification rules for the provider of a retailing market, to manually modify the taxonomy. Another approach in Lin and Hong (2008) focuses on the database development to support marketing managers when developing taxonomies for new e-catalogs. Their database consists of different components storing information about the customers, the products, and the taxonomies. They are utilizing a mining system to collect transaction data for analyzing consumer preferences and to finally help the expert to create new taxonomies. An e-catalog management system helping the experts and customers to create, update, and customize their individual taxonomy, was presented by Farsani and Nematbakhsh (2007). Their system requires feedback from the user in the form of keywords to analyze the customers' labeling preferences for making conceptual recommendations. An approach reordering search results according to users interest was presented in Fathy et al. (2014). Their system is utilizing a concept-based user profile to learn customers preferences and to prune the taxonomy based on a given keyword. A knowledge discovery framework for the construction of personalized web directories was introduced by Pierrakos and Paliouras (2010). The system is exploiting the users browsing behavior throughout the web with thematic information from the web directories. A personalized ontology model requiring rich semantics is presented in Tao et al. (2011). The system aims to learn user profiles from a world knowledge base and from a local instance repository. However, as all existing approaches are requiring human effort in the form of keywords or feedback, or require rich semantics in the form of more extensive ontologies, which are not provided in ordinary taxonomies used in e-catalogs, the creation of personalized directories is sill too time- and cost-intensive for high traffic retailing markets. Furthermore, a pragmatic approach, like in the other above-mentioned research areas, would lose the flexibility of the e-catalog when it is distributed over different cross-media channels (e.g. printed, digital, or in e-commerce). In an imposed (e)-catalog, for example, the customer should still be able to navigate through the entire product range structured with the taxonomy. Otherwise, there is a high chance of possible reduction in order.

During the last years, logic programming has turned out to be the most efficient technique for dealing with the above-mentioned challenges. In logic programming the taxonomies are not structured inside a database, but in the form of a knowledge base consisting of facts, rules, as well as queries. Through this, this programming paradigm provides multiple benefits for knowledge management in e-business, especially for taxonomical engineering (Gomez-Perez et al. (2006)). On the one hand, as logic approaches are geared to deal efficiently with larger sets of concepts, such techniques can deal with very large taxonomies, as well as with small business taxonomies. On the other hand, as logic approaches are exploiting the reasoning capabilities for automating tasks, this paradigm perfectly deals with tasks concerning the semantics of e-commerce sites. And furthermore, as logic programming is cross-platform and database-independent, the frameworks implemented with this technique can be applied to all recently available standalone e-commerce applications, as well as on distributed information management systems. This benefits have been exploited across various applications in e-business. For example, the authors in Sabater-Mir et al. (2013) are proposing a cognitive logic programming based architecture to personalize recommendations in e-commerce. Hereby, the benefits are used to learn the strength and weakness of existing recommenders to provide more reliable and trustful recommendations. The authors in Ostermayer and Seipel (2012) are using logic programming to more efficiently control complex business rules for knowledge engineering. The comparison with other programming techniques has turned out that logical programming is more beneficial for business rules then the most comprehensive business rule manager Drools that is programmed using Java. And finally, logic programming has proven to improve semantic web applications. The authors in Conen and Klapsing (2001) for example are presenting a logic-based tool named RDF Schema Explorer, which can parse, validate, query and extend RDF Schemata over different collaboration-driven application domains.

To provide a fully-automatic solution for creating personalized directories according to customers preferences, the expert system *TaxoPublish* is presented. The proposed system is performing fully-automatically personalization by combining recommendation techniques with taxonomic operations to reduce the taxonomy overload. The preference analysis component, as well as the component for creating the personalized directories are implemented in logic programming, to be exandable and database independent for all recently available e-business scenarios. In the end, *TaxoPublish* provides three contributions to the field of expert systems:

- Firstly, it provides the first methodology to analyze taxonomy overload, as well as the desired reduction of the overload. This methodology can be applied across all domains using taxonomies, and will play a crucial role in domains dealing with multi-channel applications and big data challenges.

- Secondly, it provides the first system that is personalizing the taxonomy and reducing the taxonomy overload in a fully-automatic manner. Through the usage of logic programming and a distributed knowledge base, the system can be applied in various e-commerce related, as well as in other related fields dealing with taxonomies (e.g. e-logistics, e-health).
- Thirdly, it provides the first solution that is not reducing the information about the domain, respectively the semantics inside the taxonomy. It does this by providing two novel concept types operating as flexible mediator concepts.

The remainder of the paper is organized as follows. Section 2 formulates the taxonomy overload problem for e-catalogs. The method of *TaxoPublish* is presented in Section 3 by detailing the included knowledge base, discussing the proposed mediator concepts, and explaining the provided taxonomic operations. Additionally, this section presents the implementation of the integrated recommender system and the technique to create personalized directories. A case study for two different customers is presented in Section 4. In Section 5, the system is evaluated against three databases representing retailing markets with different taxonomies, goods, and customer behaviour. The work concludes in Section 6.

2. Problem formulation

E-catalogs are collections of goods, which are utilizing taxonomies to model a field of interest in a formal way (Guarino et al. (2009), see Figure 1(a)). A *Taxonomy* (Θ) is a hierarchy of objects with similar properties (Raunich and Rahm (2012)), defined as technical terms representing domains (Pazienza et al. (2005)) with (see Equation 1):

$$\Theta = (\{\Phi\}, \{\Lambda\}), \tag{1}$$

where Φ is a partially ordered set of concepts, and Λ is a set of edges connecting concepts. The edges between the concepts represent the hierarchical relationships inside the taxonomy. For example, a taxonomy consisting of three hierarchically ordered levels utilizes a root concept as the most general concept, different super concepts detailing the root concept, and sub concepts detailing the super concept, which is, in turn, a sub concept of the root concept (see Figure 1(b)). A *Sub Concept D*, formally $subof$, is a less generalized concept of B, as given in Equation (2), if:

$$D = subof(B) :\Leftrightarrow (D \subset B) \wedge ((D \wedge B) \in \Theta), \tag{2}$$

where D and B are two concepts of taxonomy Θ. A *Super Concept* B, formally $superof$, is a more generalized concept of D, as given in Equation (3), if:

$$B = superof(D) :\Leftrightarrow D = subof(B). \tag{3}$$

A *Sibling Concept* E, formally $sibof$, is the relationship between two concepts sharing the same super concept, as given in Equation (4), if:

$$E = sibof(D) :\Leftrightarrow (E \wedge D) = subof(B). \tag{4}$$

A *Root Concept* A, formally $rootof$, is a concept that has no super concept, as given in Equation (5), in which:

$$A = rootof(\Theta) :\Leftrightarrow \nexists superof(A). \tag{5}$$

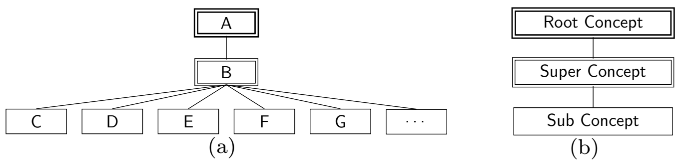


Fig. 1. The hierarchical structure of a taxonomy: a) The sample taxonomy including seven edges to connect the eight concepts; b) The different concept types inside the taxonomy.

Taxonomies are generated through the provider, or by referring to a standard taxonomy, e.g. the *North American Product Classification System*[3] (NAPCS) (Donglin et al. (2010); Schulten et al. (2001)). However, as recent applications do not provide the possibility to automatically restructure the taxonomy according to customers preferences, the taxonomy suffers from two problems, formulated as *Taxonomy Overload*.

[3]http://www.census.gov/eos/www/napcs

- Firstly, because of customers have individual and over the time changing requirements, they are affected with a number of irrelevant concepts (Δ), indicated with (see Equation 6):

$$\Delta = |\Phi_\psi| - |\Phi_\omega|, \tag{6}$$

where Φ_ψ is the set of all most detailing concepts, and Φ_ω only includes the actually required most detailing concepts. Both sets can be formulated in a taxonomy consisting of three levels as (see Equations 7 and 8):

$$\Phi_\psi = \{\Gamma|(\Gamma = subof(\Pi)) \wedge (\Pi \neq rootof(\Phi))\}; \tag{7}$$

$$\Phi_\omega = \{\Gamma|(\Gamma \in \Phi_\psi) \wedge (\Gamma \in \Sigma)\}, \tag{8}$$

where Γ and Π are concepts of Θ. Γ can be included in the considered customer profile because it is the concept categorizing an instance available for order (e.g. product). The user profile can be captured in a three-tuple (see Equation 9):

$$\Sigma = \{\zeta, \eta, \Gamma\}, \tag{9}$$

detailing the customer with an identifier ζ, and an identifier representing each purchasing process (epoch) with η.

- Secondly, as the sibling concepts are not flexible, high and low preferred sub concepts are assigned with the same semantic context weight. Thus, customers always have to filter for the required concept detailing a common super concept. Formally, the distraction can be reduced with Ξ (see Equation 10):

$$\Xi = ||\Phi_\vartheta| - |\Phi_\theta||, \tag{10}$$

where Φ_ϑ is the set of sibling concepts for $\Gamma \in \Sigma$, with (see Equation 11):

$$\Phi_\vartheta = \{\Lambda|\Lambda sibof(\Gamma) \wedge \notin \Sigma\}, \tag{11}$$

$$\Phi_\theta = \{\Lambda|\Lambda sibof(\Gamma)\}, \tag{12}$$

where Λ is another sub concept of Π.

3. Proposed system TaxoPublish

The proposed system is performing on two new defined types of taxonomic concepts, allowing to analyze dynamic preferences and to perform different taxonomy operations. Both types are not shown in the initial taxonomy but are used to personalize/reduce the taxonomy according to customers' preferences without excluding any part of the domain. *TaxoPublish* is implemented with logic programming. Through this, the proposed architecture is flexible to be used in different relational databases (e.g. HHGMultistore[4], or OpenCart[5]), or hierarchical databases (e.g. wooCommerce[6], or Arcavias[7]). Furthermore, the proposed implementation is extendable, independent, and understandable by the help of a compressed data structure.

3.1. TaxoPublish method

As a taxonomy already provides hierarchical relationships in the form of different concept types, the existing correlations can be utilized to create new correspondences. In this paper, two new terms are presented: *Taxonomic Dependencies* and *Taxonomic Bindings*. Those can be seen as a flexible mediator between a super and its sub concepts. In *TaxoPublish*, the dependencies are used to analyze preferences between more related sibling concepts. The bindings are required to correct the semantics of superordinated concepts depending on the operations performed on the dependencies.

3.1.1. Knowledge base

In logic programming, each program is structured as a sequence of clauses, called knowledge base (Bramer (2014)). It uses predicates in the form of facts for expressing data entities, and rules for defining relationships between the facts (Merrit (2000)). Facts consist of predicates standing before the clinches, and arguments standing between the clinches (e.g. $fact(argument1, argument2)$) (Bramer (2014)). The short form of predicates adds the number of arguments behind a horizontal line (e.g., $fact/3$). Arguments can either be atoms (e.g., *company*), numbers (e.g., 1), or variables (e.g., CCI). A rule describes a collection of requirements that have to be fulfilled to unify a query

[4]https://www.hhg-multistore.com/
[5]http://www.opencart.com/
[6]http://www.woothemes.com/woocommerce/
[7]http://www.arcavias.com/

(*true*), otherwise it fails (*false*). Each rule consists of a header and a body, which are separated with the if character ($:-$). The header is also a predicate including arguments in the form of variables. A special form of the variable is called anonymous variable. It is written as underscore (_) and has no effect on other facts. The body itself can consist of other rules, facts, and regular expressions.

The knowledge base utilized in *TaxoPublish* defines the minimal number of facts being required to analyze customer preferences for a B2B retailing market. It combines the knowledge provided through two information management systems, namely a Customer Relationship Management (CRM) system, and a PIM system. A CRM system as a repository of customer information (Phan and Vogel (2010)), e.g. the customer taxonomy. The Product Information Management (PIM) component to concern the recording of product relevant content, e.g. the product taxonomy. Six facts are required for representing the CRM knowledge, and three facts are necessary to represent the PIM system:

- *kcco*(CCI, CCL, CSI) represents the companies/customers with three arguments: CCI as identifier, CCL for capturing the companies name, and CSI for the unification of the company to a related sub concept of the CRM taxonomy.

- *kcpc*(CPI, CCI) connects the persons being responsible for the companies' purchases with two arguments: the identifier of a person CPI, and CCI.

- *kcse*(CEI, CEL) captures the super concepts, and *kcss*(CSI, CSL, CEI) represents the sub concepts of the CRM taxonomy. Both integrate an identifier (CEI, and CSI), and a compound term for capturing the label of the concept (CEL, respectively CSL). The relationship between both is realized with adding CEI as the third argument to *kcss*/3.

- *kcod(CFI,CRI)* and *kcoh*(CFI, CPI) express the companies' orders. Both use in common three arguments: CFI to identify the order, CRI to express the products, and CPI to unify with the customer.

- *kpgr*(PGI, PGL) and *kpcl*(PCI, PCL, PGI) reveal to the taxonomy of products, the PIM taxonomy. The former is used to capture its super concepts, and the later is used to express the sub concepts. Its arguments are equal to the structure used in *kcse*/2, and *kcss*/3: PGI and PLI are used for the unification between each other, PGL and PLL are utilized for assigning the labels of the concepts.

- *kppr(CRI,CRL,PLI)* is used for structuring the products, the actual documents of the PIM knowledge base. It uses a unique identifier CRI, an argument for a name CRL, and PLI as third argument to unify with the corresponding sub concept.

3.1.2. Taxonomic dependencies

The term of *Taxonomic Dependency* relies on the fact that sub concepts can have more complex relationships between each other than only connecting those with a common super concept (see Figures 2(a) and 2(b)). The dependence between siblings can vary from being more likely an antonym (e.g., “Coffee” and “Beer”) or synonym (e.g., “Coffee” and “Tea”), and being a hypernym (“Hot Drink”) or hyponym (“Decaffeinated”) to other sibling concepts, e.g. used in the semantic lexicon *WordNet*[8]. Taxonomic dependencies offer the possibility to execute taxonomic operations, as they do not over-specialize on the single sub concept but also do not under-specify on the common super concept.

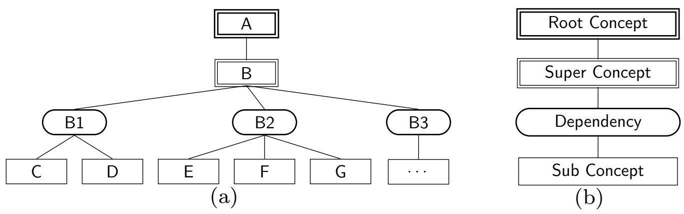


Fig. 2. The hierarchical structure of a taxonomy with dependencies: a) The sample taxonomy including eight concepts and three dependencies ; b) The different concept types inside the taxonomy.

Formally, a *Taxonomic Dependency* $B1$ (shortened as $depof$), is a mediator between a super concept B and a set of sub concepts in Υ, as given in Equation 13, if:

$$B1 = depof(B, \Upsilon) :\Leftrightarrow \forall\chi((\chi \in \Upsilon) :\Leftrightarrow \rho > \tau), \tag{13}$$

where ρ is a verified further semantic relationship (e.g. hypernym) between the sibling concepts $\in \Upsilon$, and τ is the threshold provided through the provider to verify the relationship between the sibling concepts. In logic programming, this notion can be implemented with two facts.

[8]https://wordnet.princeton.edu/

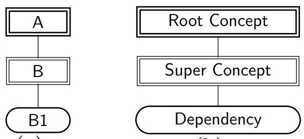
- *kodl*(ODI, ODL) captures the name of the dependency with two arguments: an identifier in ODI, and ODL to assign a label to the dependency.

- *kodd*(PGI, ODI, PLI) connects the dependency to its super concept and the sub concepts. Its structure is equal to the structure inside the taxonomy.

3.1.3. Taxonomic operations

Because of the dependencies are not shown in the initial taxonomy, those offer the possibility for various taxonomic operations. Each operation is described with three arguments defining different levels of the taxonomy. The performed operations depend on preferences analyzed through an integrated recommender system. It measures the state of past-term preferences ($RVS = \{low; middle; high\}$), and the state of future-term preferences ($EVS = RVS$), to indicate the companies' final preferences for the upcoming epoch (HVS). The operator can define, which combinations are necessary to result in the final state of preference in $kxst(RVS, EVS, HVS)$, e.g. $kxst(middle, low, low)$ to put more priority on the future-term analysis. Recently, three operations on dependencies are considered to reduce the taxonomy overload:

- *Bundling Operation* combines low preferred sibling concepts to a single dependency (see Figures 3(a) and 3(b)). It reduces the taxonomy overload by the number of siblings belonging to the same super concept but not to the same dependency (in $tub(CCI, PGI, ODI)$).



(a) (b)

Fig. 3. The modification of the taxonomy with the bundling operation performed on a taxonomy dependency: a) Illustrative Example; b) Relationships inside the taxonomy.

- *Splitting Operation* divides middle preferred dependencies in single sub concepts (see Figures 4(a) and 4(b)). Thus, the customers are still able to reach possibly required concepts of a super concept with the identical semantic context (in $tus(CCI, PGI, PLI)$).

- *Moving Operation* puts the dependency to the level of a super concept (see Figures 5(a) and 5(b)). It significantly increases the semantic context weight of highly required dependencies,

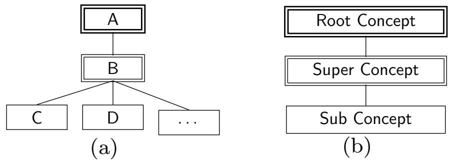


Fig. 4. The modification of the taxonomy with the splitting operation performed on a taxonomy dependency: a) Illustrative Example; b) Relationships inside the taxonomy.

respectively its assigned sub concepts (in $tuv(CCI, ODI, PLI)$).

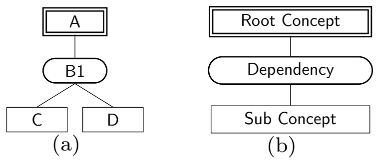


Fig. 5. The modification of the taxonomy with the moving operation performed on a taxonomy dependency: a) Illustrative Example; b) Relationships inside the taxonomy.

3.1.4. Taxonomic bindings

Executing the moving operation on dependencies effects that the super concept is semantically reduced by the number of moved dependencies. Thus, the remaining super concept only includes a subset of elements and its initial label would be misleading. For that reason, the term of *Taxonomic Binding* is defined representing a sibling of a super concept including multiple dependencies except the moved ones (see Figures 6(a) and 6(b)) with (see Equation 14):

$$B12 = bindof(B1, B2) :\Leftrightarrow \{B1, B2, B3\} - \{B3\} \tag{14}$$

where $B1$, $B2$, and $B3$ are dependencies of the super concept B, but $B3$ is modified with the moving operation. In logic programming, bindings can be depicted with utilizing two clauses:

- $kobl(OBI, OBL)$ carries the name of the binding in (OBL) along with an identifier (OBI).
- $kobb(OBI, ODI)$ connects the binding with the dependencies captured in *kodd/3*.

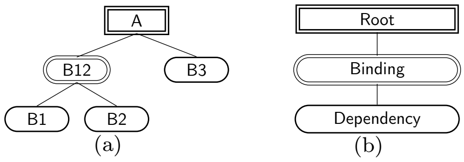


Fig. 6. The hierarchical structure of a taxonomy with bindings based on dependencies: a) A sample taxonomy including three dependencies; b) The modification with a binding applied to two dependencies; c) The structure of the taxonomy including bindings (for the first two levels).

3.2. TaxoPublish implementation

User profiles can be utilized for analyzing dynamic customer interests. Those can be captured by either explicitly or implicitly, regarding Agichtein et al. (2006). By the explicit approach, users proactively communicate information to the system (Calegari and Pasi (2013)). The following components are needed to analyze the preferences by the implicit approach and to modify the taxonomies automatically.

- *Forecasting requirements* to forecast the customers' demand for the next shopping process. It utilizes three steps without requiring customers' or the providers' feedback:
  - *Past-term analysis* for deriving the companies' long- and short-term preferences. The analysis is based on the products ordered in different epochs.
  - *Future-term analysis* to forecast requirements. The CRM classification is exploited to filter for customers sharing the same super- and/or sub concept.
  - *Hybrid aggregation* combines past- and future-term results to analyze the customers' final state of preference. Its result affects one of the three operations.
- *Personalizing taxonomy* to modify the taxonomy automatically based on the results of the preference analysis:
  - *Identification* to identify one set including moved dependencies and another set illustrating the remaining dependencies of an affected super concept.
  - *Binding all/none* satisfies if one of the both indicated sets is empty. The super concept can either be replaced completely or can remain completely.

– *Binding some* unify if both sets are not empty. The label of the super concept is replaced by a binding. The unification is based on the remaining dependencies.

– *Output* performs for all super concepts, the three previous mentioned steps iteratively. The rule satisfies if for all super concepts a solution is found.

3.2.1. Past-term analysis

Past-Term preferences are interests combining long- and short-term preferences. Long-term preferences are interests that have been shown in the more distant past (Shen et al. (2005)). Short-term preferences happened in a more recent past (Teevan et al. (2010)). The analysis is performed with the predicate $rs/3$ (see Listing 1).

```
rs(CCI,ODI,RVS):-
  rsp(CCI,ODI,RVW),
  rst(CCI,RHD,RHH),
  ( (RVW>=RHH,RVS=high);
    (RVW>=RHD,RVW<RHH,RVS=middle);
    (RVW<RHD,RVS=low)).
```

Listing 1: rs/3

The rule $rs/3$ includes a comparison between the value of preference (RVW) and two thresholds (RHD, and RHH) to indicate the corresponding status of preference (RVS). To do this, $rs/3$ recalls four rules against each other:

1. Dividing the past-term preferences into different rated epochs reveal to the fact that user interests usually change over time. The division into different epochs is performed with $rsi/5$ (see Listing 2).

```
rsi(CCI,ROI,ROW,RVLP,RVLN):-
  rsic(CCI,CFI,PLL),
  rsie(CCI,CFI,ROI,ROW),
  findall(ODI,(member(PLI,PLL),kodd(_,ODI,PLI)),RVLP),
  findall(ODI,(kodl(ODI,_),not(member(ODI,RVLP))),RVLN).
```

Listing 2: rsi/5

It firstly identifies the different orders for a customer, which are reduced to epochs (ROI). For each epoch, a rate (ROW) is assigned in the rule $rsie/4$. This weight is based on the time function introduced in Ding and Li (2005). In contrast to other works, their method offers the possibility to include a domain-specific variable to decrease the preference value for less recent epochs as follows (see Equation 15):

$$ROW_{ROI} = e^{(-((1/(RON*VOW))*(1/2))*(RON-ROI))}, \tag{15}$$

where RON is the number of epochs a customer has shown, and ROI is the considered epoch. It can be adopted through the operator inside the fact $rsdr/1$. For each epoch, a list of preferred dependencies $RVLP$, and a list of not preferred dependencies $RVLN$ is queried in $rsic/3$.

2. The similarity measure in $rss/2$ states how loyal the customer stays its preferences (see Listing 3).

```
rss(CCI,RYW):-
  rsio(CCI,_,CFN),
  ( ( CFN>1,rssk(CCI,RYWK),rssy(CCI,RYWJ),
      RYW is (0.5*RYWJ)+((1-0.5)*RYWK));
    ( CFN=<1,RYQ is 1)).
```

Listing 3: rss/3

The predicate utilizes two foregone rules: $rssk/2$, and $rssy/2$ to estimate an epoch-epoch similarity, in accordance to the user-user similarity introduced in Li et al. (2012). It combines the average mean of *Jaccard* similarities between all epochs and its *Cosine* similarity. In *TaxoPublish*, both measurements are based on $RVLN$.

3. As next, the value of preference (RVW) is assigned to each dependency in $rsp/3$ (see Listing 4).

```
rsp(CCI,ODI,RVW):-
  kodl(ODI,_),
  findall(ROW,(rsi(CCI,_,ROW,RVLP,_),member(ODI,RVLP)),ROL),
  sumlist(LTR,ROU),length(LTR,RON),
  ( (RON > 0,RVW is ^((ROU/RON),(1/RON)));
    (RON = 0,RVW is 0)).
```

Listing 4: rsp/3

Its weight is calculated with the average mean of epoch rates assigned to one dependency. The result is normalized by the number of different epochs shown by the customer. To do this, $rsp/3$ includes $rsi/5$ inside the built-in predicate $findall/3$ to fill a list including all epoch rates (ROL).

4. Because of three status of preferences are used to highlight the different interests on dependencies, two thresholds are queried in $rst/3$ (see Listing 5): RHD for middle preference, and RHH for high preference.

```
rst(CCI,RHD,RHH):-
  rstt(CCI,RHDI,RHHI),rstl(CCI,RZW),
  RDH is RHDI / RZW,RHH is RHHI / RZW.
```

Listing 5: rst/3

The rule performs a comparison between the predicates $rstt/4$ and $rstl/3$. The former exists to measure the intermediate thresholds ($RHDI$, respectively $RHHI$). The later assigns a tolerance value to the thresholds. The measurement of the intermediate thresholds is based on the range of preference values (RVW) of the customer, respectively its average mean. A tolerance value is added (RZW in $rstl/2$), to minimise the influence of number of orders on the truth value, as explained in previous section. It normalizes the lower similarity value of the customers' having more epochs (see Equation 16):

$$RZW = \frac{100 + e^{2\frac{RON}{RONV}}}{100}, \tag{16}$$

where RON is the number of epochs of the considered customer, and $RONV$ is the average mean of the number of epochs for all customers.

3.2.2. Future-term analysis

Future-term preferences are interests carried out after the customers' most recent epoch. Its analysis is based on the most recent preferences of similar customers happened after the most recent epoch of the considered customer. The resulting preferences are weighted with the semantic context weight of the compared customers to infer the similarity between the customers (see Viswanathan and Krishnamurthi (2012) for further details). Consequently, this type of analysis results the expected preferences for the customers' upcoming epoch with the predicate $rv/3$ (see Listing 6).

```
rv(CCI,ODI,EVS):-
  rvp(CCI,ODI,EVW),rvt(CCI,EHD,EHH),
  ( (EVW>=EHH,EVS=high);
    (EVW>=EHD,EVW<EHH,EVS=middle);
    (EVW<EHD,EVS=low)).
```

Listing 6: rv/3

The preferences are determined with exploiting the CRM taxonomy, namely by searching for users sharing the same sub concept, or at least the same super concept. The resulted preference values, as similar to the past-term analysis, are compared with two thresholds. To do this, $rv/3$, respectively the rules $rvp/3$ and $rvt/3$ inside its body, recall five rules against each other:

1. The result of $rve/3$ includes the customers being similar to the considered customer in ECL (see Listing 7).

```
rve(CCI,ECL,EXW):-
  rvil(CCI,CFX),
  ( kcco(CCI,_,CSI),
    findall(ECI,(kcco(ECI,_,CSI),rvil(ECI,EFX),EFX>CFX),ECL),
    length(ECL,ECN),ECN>0,EXW is 1,!);
  ( kcco(CCI,_,CSI),rvim(CCI,CEI,CSI),
    rvic(CEI,CSN),rvil(ECI,EFX),
    findall(ECI,(rvim(CCI,CEI,_),rvil(ECI,EFX),EFX>CFX),ECL),
    length(ECL,ECN),ECN>0,EXW is 1 / sqrt(CSN)).
```

Listing 7: rve/3

The first sequence aims to detect customers sharing the same sub concept of the CRM taxonomy. If this sequence fails, the second sequence searches for customers sharing the same super concept. Hereby, the value describing the semantic closeness (EXM) is taken into account. It decreases by the number of edges (CSN in $rvic/2$) belonging to the super concept.

2. For all elements of ECL, the latest preferences are analyzed in $rvi/3$, respectively its sub ordinated clause $rvio/3$ (see Listing 8). The latter results an unsorted list of dependencies inside the second argument ODL.

```
rvi(CCI,ODI,EKN):-
  kodl(ODI,_),rvio(CCI,ODL,_),
  findall(ODI,member(ODI,ODL),ODK),
  length(ODK,EKN),EKN>0.
```

Listing 8: rvi/3

3. Similar to $rst/3$, two thresholds are computed in $rvt/3$ (see Listing 9). Both thresholds emerge from the range (EKG) of occurrences shown on taxonomic dependencies.

```
rvt(CCI,EHD,EHH):-
  findall(EKN,rvi(CCI,_,EKN),EKL),
  min_list(EKL,EKM),max_list(EKL,EKX),
  EKG is EKX-EKM,
  EHD is EKX-((EKG/3)*2),
  EHH is EKX-(EKG/3).
```

Listing 9: rvt/3

For example, a sub concept ordering many products has a higher threshold compared to a sub concept ordering fewer products. This results that for the creation of personalized directories, only the most important dependencies are assigned with high preference.

4. To avoid overemphasizing of the results coming from $rv/3$, a normalization is applied with $rvc/2$ (Listing 10).

```
rvc(CCI,ETW):-
  rvtv(CCI,EWW),rvie(CCI,_,EXW),rvcw(VWW,VXW),
  VTU is VWW + VXW,ETW is (VWW * EWW) + (VXW * EXW)) / VTU.
```

Listing 10: rvc/2

This predicate combines the semantic context weight along with a variable called overload (EWW inside $rvtv/2$). The overload reduces the number of preferences for a list of customers by the number of preferences per epoch of the considered customer. It ensures that only very strong preferred dependencies have an influence on the personalized taxonomies, with (see Equation 17):

$$EWW = \sqrt[RONV]{\frac{1}{\sqrt{|RONV - EONV| + 1}}}, \tag{17}$$

where $RONV$ is the average mean of the number of dependencies preferred in epochs, and $EONV$ is the average mean of the number of dependencies preferred in epochs through similar customers.

5. Finally, the occurrences for dependencies are quantified with the above-mentioned truth value to result weighted preferences (EVW in $rvp/3$, see Listing 11).

```
rvp(CCI,ODI,EVW):-
  ( kodl(ODI,_),rvi(CCI,ODI,EKN),
    rvc(CCI,ETW),EVW is EKN * ETW);
  ( kodl(ODI,_),not(rvi(CCI,ODI,_)),EVW is 0).
```

Listing 11: rvp/3

Combining past-term and future-term preferences is realized in $r/3$. It represents a hybrid analysis of distant and expected interests and states the companies final state of preference and is responsible to satisfy the taxonomic operations. The aggregation is realized with the predicate $kxst/3$. It utilizes the state of past-term preferences ($RVS = \{low; middle; high\}$), and the state of future-term preferences ($EVS = \{low; middle; high\}$) to outline the companies final preferences for the upcoming epoch ($HVS = \{low; middle; high\}$). This combination allows to be flexible over different channels. For example, a multi-channel retailer can provide one combination for the e-catalog, and another combination for the printed catalog.

3.2.3. Hybrid aggregation

Combining past-term and future-term preferences is realized in $r/3$ (see Listing 12).

```
r(CCI,ODI,HVS):-
  kcco(CCI,_,_),kodl(ODI,_),
  ( rv(CCI,ODI,EVS),rs(CCI,ODI,RVS),kxst(EVS,RVS,HVS));
  ( not(rv(CCI,ODI,_)),rs(CCI,ODI,RVS)).
```

Listing 12: r/3

3.2.4. Identifying concepts

Identifying reduced super concepts aims in finding two unique sets of dependencies belonging to one identical super concept of the PIM taxonomy (in $ti/4$, see Listing 13). The first set includes only high preferred dependencies (moving operation), whereas the second set includes only middle (splitt operation) and low (bundling operation) preferred dependencies.

```
ti(CCI,PGI,BIOR,BIOM):-
  tig(PGI,ODO),
  tid(CCI,PGI,BIOR),
  subtract(ODO,BIOR,BIOM).
```

Listing 13: ti/4

The rule includes two foregone unifications:

1. To create the first required set (ODO) for *ti*/4 including all dependencies sharing a common super concept, the clause *tig*/2 is recalled (Listing 14).

```
tig(PGI,ODO):-
  kpgr(PGI,_),
  findall(ODI,kodd(PGI,ODI,_),ODL),list_to_set(ODL,ODO).
```

Listing 14: tig/2

2. To indicate a set including all reduced dependencies, the rule *tid*/3 is performed (see Listing 15). Its body relies on the moving operation performed with *tuv*/3.

```
tid(CCI,PGI,BIOR):-
  kpgr(PGI,_),
  findall(ODI,(kodd(PGI,ODI,_),tuv(CCI,ODI,_)),BILR),
  list_to_set(BILR,BIOR).
```

Listing 15: tid/3

3.2.5. Binding all/none

In the case of all dependencies of a super concept are highly preferred (moving operation), no binding is required. All dependencies are labeled with itself, treated in *ta*/3 (see Listing 16). It results the final paths consisting of a dependency as a super concept and its sub concepts.

```
ta(CCI,ODI,PLI):-
  kpgr(PGI,_),ti(CCI,PGI,BIOR,BIOM),
  length(BIOM,BINM),length(BIOR,BINR),
  BINM=0,BINR>0,member(ODI,BIOR),tuv(CCI,ODI,PLI).
```

Listing 16: ta/3

Similar, if not any dependency of a super concept is moved to the higher level, no binding is required. The label of the super concept can remain, treated in *tn*/3 (see Listing 17).

```
tn(CCI,PGI,ODI):-
  kpgr(PGI,_),ti(CCI,PGI,BIOR,BIOM),
  length(BIOM,BINM),length(BIOR,BINR),
  BINR=0,BINM>0,member(ODI,BIOM),tub(CCI,PGI,ODI).
tn(CCI,PGI,PLI):-
  kpgr(PGI,_),ti(CCI,PGI,BIOR,BIOM),
  length(BIOM,BINM),length(BIOR,BINR),
  BINR=0,BINM>0,member(ODI,BIOM),tus(CCI,PGI,PLI).
```

Listing 17: tn/3

3.2.6. Binding some

The last case requires the label of the binding, because of only a subset of dependencies is moved to a super concept. Thus, the customer would expect products, which are not included anymore in the initial super concept. The relabeling is performed with the clause $ts/3$ (see Listing 18).

```
ts(CCI,BAVA,BAVB):-
  tsm(CCI,BAVA,BAVB);
  tso(CCI,BAVA,BAVB);
  tsv(CCI,BAVA,BAVB).
```

Listing 18: ts/3

$ts/3$ combines three sequences inside its body:

1. The rule $tsm/3$ (see Listing 19) renames a super concept with the binding associated to the remaining set of dependencies. Its logical expression includes the clause $tsxb/2$. It queries for a list including a list of all possible bindings including the associated dependencies. This list is compared with the built-in predicate $subtract/3$. Its result is the identifier TBB of $txsb/2$ where the subtraction results in an empty list. Again two different bodies are used to treat the clauses of $tub/3$ and $tus/3$.

```
tsm(CCI,OBI,ODI):-
  ti(CCI,PGI,BIOR,BIOM),
  length(BIOM,BINM),length(BIOR,BINR),
  BINR>0,BINM>1,tsxb(OBI,ODL),
  subtract(BIOM,ODL,BIOA),
  length(BIOA,BINA),
  BINA = 0,tub(CCI,PGI,ODI).
tsm(CCI,OBI,L2):-
  ti(CCI,PGI,BIOR,BIOM),
  length(BIOM,BINM),length(BIOR,BINR),
  BINR>0,BINM>1,tsxb(OBI,ODL),
  subtract(BIOM,ODL,BIOA),
  length(BIOA,BINA),
  BINA = 0,tus(CCI,PGI,PLI).
```

Listing 19: tsm/3

2. Another rule *tso*/3 is required when the remaining super concept only consists of one dependency (see Listing 20). Therefore, the sub concept is renamed by exploiting the clause *kodd*/3, either in *tus*/3, or in *tub*/3. Besides the different bodies, both predicates further differ in their third argument. The built-in predicate *member*/2 is used to replace the initial super concept with the associated dependency in the case of *tus*/3, as it queries multiple sub concepts belonging to a dependency. In the case of the bundling operation, the third argument is duplicated with the help of the second argument.

```
tso(CCI,ODI,PLI):-
  ti(CCI,PGI,BIOR,BIOM),
  length(BIOM,BINM),length(BIOR,BINR),
  BINR>0,BINM=1,member(ODI,BIOM),
  kodd(PGI,ODI,PLI),tus(CCI,PGI,PLI).
tso(CCI,ODI,ODI):-
  ti(CCI,PGI,BIOR,BIOM),
  length(BIOM,BINM),length(BIOR,BINR),
  BINR>0,BINM=1,member(ODI,BIOM),
  kodd(PGI,ODI,_),tub(CCI,PGI,ODI).
```

Listing 20: tso/3

3. The last predicate of *tsv*/3 treats the high preferred dependencies of a reduced super concept (see Listing 21).

```
tsv(TY,ODI,PLI):-
  kpgr(PGI,_),ti(CCI,PGI,BIOR,BIOM),
  length(BIOM,BINM),length(BIOR,BINR),
  BINR>0,BINM>0,
  member(ODI,BIOR),tuv(CCI,ODI,PLI).
```

Listing 21: tsv/3

3.2.7. Output

The identification of super concepts is performed for each fact iteratively in $t/3$ (see Listing 22). It satisfies if for all super concepts a solution is found.

```
t(CCI,BAEA,BAEB):-
  distinct([CCI,BAEA,BAEB],
  ( ta(CCI,BAEA,BAEB);tn(CCI,BAEA,BAEB);ts(CCI,BAEA,BAEB))).
```

Listing 22: t/3

4. Case study

As an illustration of *TaxoPublish*, let us consider the following scenario: The marketing expert of the retailing market *Northwind*[9] wants to personalize taxonomies for two customers: For the seafood market “Blauer See Delikatessen” ($CCI = 6$), and for the restaurant “Godos Cocina Tpica” ($CCI = 30$). The relational database consists of a CRM component and different tables representing a PIM system. Both components are utilizing taxonomies for the classification of the available products, and to semantically group the different customers. Let us assume that the provider has defined the taxonomic dependencies and bindings for the PIM taxonomy, as given in Figures 7 and 8 for the complete product taxonomy. Note that a binding is required if the super concept consists of more than two dependencies. In our example for the super concepts “Beverages” and “Condiments”. The variable affecting the computation of epoch-rates is ignored ($VOW = 1$), analogues to the variables affecting the normalization of the future-term analysis ($VWW = VXW = 1$), to allow the most illustrative example for the case study at hand. The operations are performed for the second level (moving operation, binding operation), and the third level (splitting operation, and

[9]https://northwinddatabase.codeplex.com/

bundling operation) of the taxonomy. Hereby, the hybrid aggregation summarized in Listing 23 is used.

```
kxst('high','high','high').
kxst('high','middle','high').
kxst('high','low','middle').
kxst('middle','high','middle').
kxst('middle','middle','middle').
kxst('middle','low','middle').
kxst('low','high','middle').
kxst('low','middle','low').
kxst('low','low','low').
```

Listing 23: kxst/3

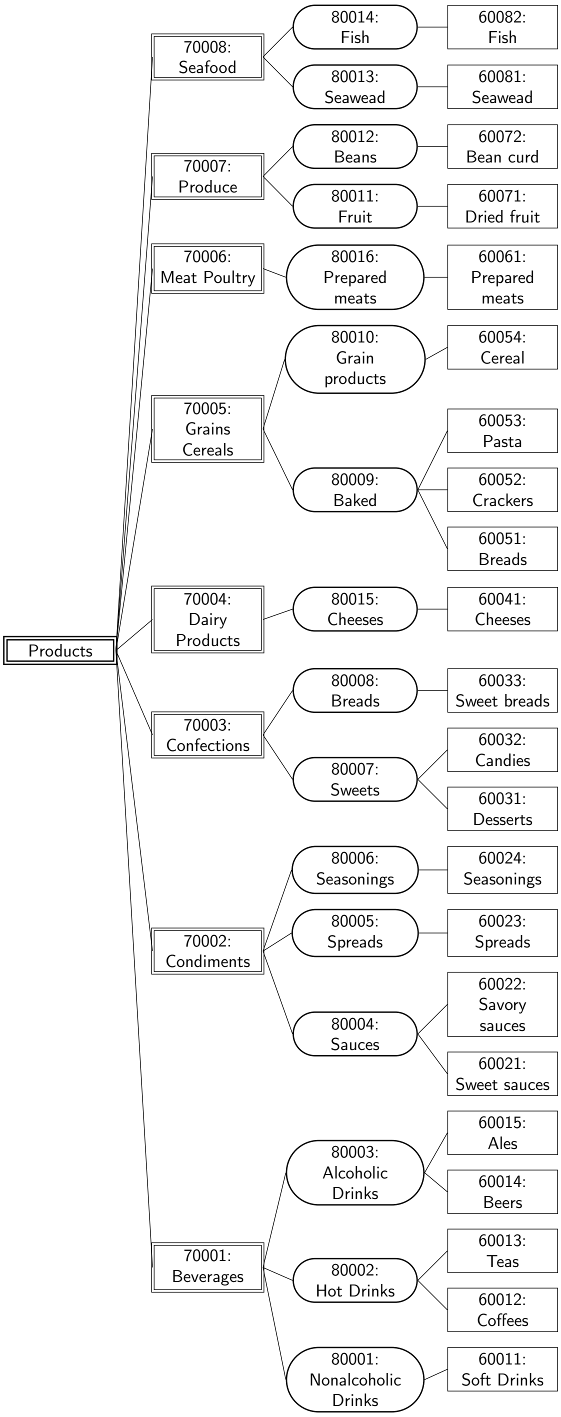


Fig. 7. The mediator concept taxonomy dependencies defined for the *Northwind* database.

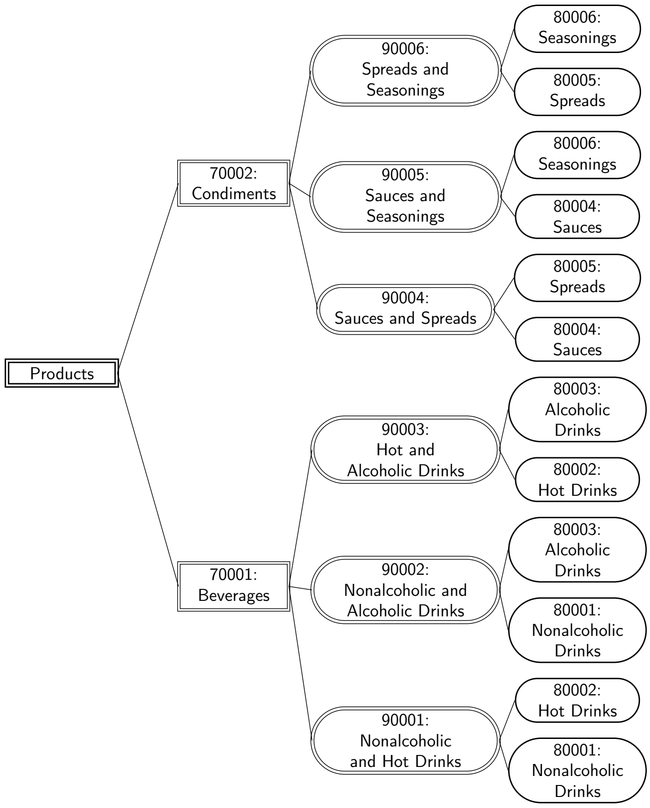


Fig. 8. The mediator concept taxonomy bindings defined for the *Northwind* database.

4.1. Forecasting requirements

The framework searches with the rule $r/3$ for the preferences shown on taxonomic dependencies to forecast the taxonomic requirements for the upcoming epoch. To do so, the system firstly identifies the past-term preferences by analyzing the customers foregone epochs. Secondly, the system is analyzing the preferences of similar customers through exploiting the CRM taxonomy.

The outcome of the first step highlights that the two investigated customers have shown a different purchasing behaviour (summarized in Table 1). The number of epochs (RON) varies from 7 ($CCI = 6$) to 10 ($CCI = 30$), which is below/above the average of all users ($RONV = 9.42$). Accordingly, each considered customer is holding unique epoch rates (ROW) for each epoch (ROI). The lists including preferred dependencies ($RVLP$) are also different when comparing both customers. On the one side, the seafood market ordered in total nine different dependencies.

However, only the dependencies “Sweets”, “Cheeses”, and “Alcoholic drinks” occur in more than one epoch. On the other side, the restaurant has ordered twelve different dependencies, but four dependencies occur in more than two epochs (“Alcoholic drinks”, “Sauces”, “Sweets”, and “Prepared meats”). Furthermore, the dependency “Alcoholic Drinks” was also ordered in the most recent epoch and thus is assigned with the highest epoch rate.

Table 1
Assigning an individual rate to more distant and more recent epochs.

CCI	ROI	ROW	$RVLP$
6	1	0.65	[80016]
	2	0.70	[80012]
	3	0.75	[80009, 80003]
	4	0.81	[80015, 80003, 80007]
	5	0.87	[80013]
	6	0.93	[80007, 80007,80011]
	7	1.00	[80007, 80015,80004]
30	1	0.64	[80013, 80004,80008]
	2	0.67	[80016, 80007,80007,80004]
	3	0.70	[80016, 80010]
	4	0.74	[80016, 80007,80010,80004]
	5	0.78	[80013]
	6	0.82	[80003, 80008]
	7	0.86	[80002,80016,80003]
	8	0.90	[80007, 80011,80016]
	9	0.95	[80001,80014,80015]
	10	1.00	[80003]

Because of the restaurant has more epochs to be compared for resulting the similarity measure, the system applies a higher tolerance value (RZW) for weighting the intermediate thresholds indicating the status of preference on a taxonomic dependency (see Table 2): $RHDI$ for middle preference, and $RHHI$ for high preference. For example, RZW is higher for the customer where $RON = 10$ than for the customer where $RON = 7$. Finally, the weighting with RWZ affects the final thresholds for each customer (RHD for middle preference, and RHH for high preference).

Table 2
Truth values and thresholds depending on the customers loyalty regarding preferences.

CCI	RYW	$RHDI$	$RHHI$	RZW	RHD	RHH
6	0.84	0.79	0.93	1.04	0.76	0.89
30	0.80	0.89	0.94	1.06	0.84	0.89

The outcome of the future-term analysis results that not for all considered customers, more recent active similar users exist. For the restaurant, a set of similar customers was found inside the same CRM sub concept, on the one hand. Moreover, the comparable companies also ordered "Cheeses" during the most recent epoch. On the other hand, for the seafood market, the system has to search on the level of the superordinated CRM concept. Because of the context weight is different when comparing companies sharing the same sub sector, than sharing the more general super concept, the semantic context has to be reduced. It decreases with the number of outgoing edges. In our example it results: $EXM = 0.71$ because the sub concept has one sibling concept. Again, the similar customers also ordered the same dependency during the most recent epoch ("Alcoholic drinks"), and through this is assigned with the highest preference value (EVW). Finally, both preference values are compared with its thresholds to indicate the final state of preference for performing the taxonomy operations (see Table 3).

Table 3
State of aggregated preferences for the different customers.

CCI	ODI	RVW	RVS	EVW	EVS	HVS
6	80001	0	low	0	low	low
	80002	0	low	0.73	low	low
	80003	0.88	middle	0.73	low	low
	80004	1.00	high	0	low	middle
	80005	0	low	0	low	low
	80006	0	low	0	low	low
	80007	0.98	high	0.73	low	middle
	80008	0	low	0	low	low
	80009	0.75	low	0	low	low
	80010	0	low	0	low	low
	80011	0.93	high	0	low	middle
	80012	0.70	low	0	low	low
	80013	0.87	middle	0	low	low
	80014	0	low	0	low	low
	80015	0.95	high	1.47	middle	middle
	80016	0.65	low	0	low	low
30	80001	0.95	high	1.75	middle	middle
	80002	0.86	middle	0	low	low
	80003	0.96	high	2.63	high	high
	80004	0.88	middle	0	low	low
	80005	0	low	1.75	middle	middle
	80006	0	low	0	low	low
	80007	0.93	high	1.75	middle	middle
	80008	0.85	middle	0	low	low
	80009	0	low	0	low	low
	80010	0.85	middle	0	low	low
	80011	0.90	high	0	low	middle
	80012	0	low	0.88	low	low
	80013	0.84	low	0	low	low
	80014	0.95	high	0	low	middle
	80015	0.95	high	1.75	middle	middle
	80016	0.95	high	0	low	middle

4.2. Personalizing taxonomy

The results of $r/3$ are now considered for building the personalized taxonomies with the predicate $t/3$. Firstly, the moving, splitting, and bundling operations are performed. For the seafood market, the dependencies "Sauces", "Sweets", "Fruit", and "Cheeses" are splitted into single sibling sub concepts, whereas the other dependencies are bundled together. For the restaurant, the dependencies "Nonalcoholic drinks", "Spreads", "Sweets", "Fruit", "Fish", "Cheeses", and "Prepared meats" remain as single sub concepts. The dependency “Alcoholic drinks” is moved to the level of a super concept inside the personalized directory for the restaurant, and its sub concepts occur as single sibling concepts.

Next, the solution searches for distinct paths based on the changed semantic compared to the initial taxonomy, depending on the foregone taxonomy operations (see Table 4). The algorithm works as follows: As for the seafood market, no dependency is moved to a higher level, the super concepts can remain (see Figure 9). In contrast, for the restaurant, high preference was indicated for the dependency “Alcoholic Drinks”. Through this, only the dependencies “Hot Drinks” and “Nonalcoholic Drinks" remain inside the super concept “Beverages”. This affects that the initial label would be misleading. For that reason, the label is changed with the label of the included dependencies captured inside the binding, named “Nonalcoholic and Hot drinks” (see Figure 10).

Table 4
Identifying reduced super concepts depending on the taxonomic operations.

CCI	*PGI*	*BIOR*	*BIOM*
6	70001	[]	[80001, 80002,80003]
	70002	[]	[80004, 80005,80006]
	70003	[]	[80007,80008]
	70004	[]	[80015]
	70005	[]	[80009, 80010]
	70006	[]	[80016]
	70007	[]	[80012, 80011]
	70008	[]	[80013, 80014]
30	70001	[80003]	[80001, 80002]
	70002	[]	[80004, 80005, 80006]
	70003	[]	[80007, 80008]
	70004	[]	[80015]
	70005	[]	[80009, 80010]
	70006	[]	[80016]
	70007	[]	[80012, 80011]
	70008	[]	[80013, 80014]

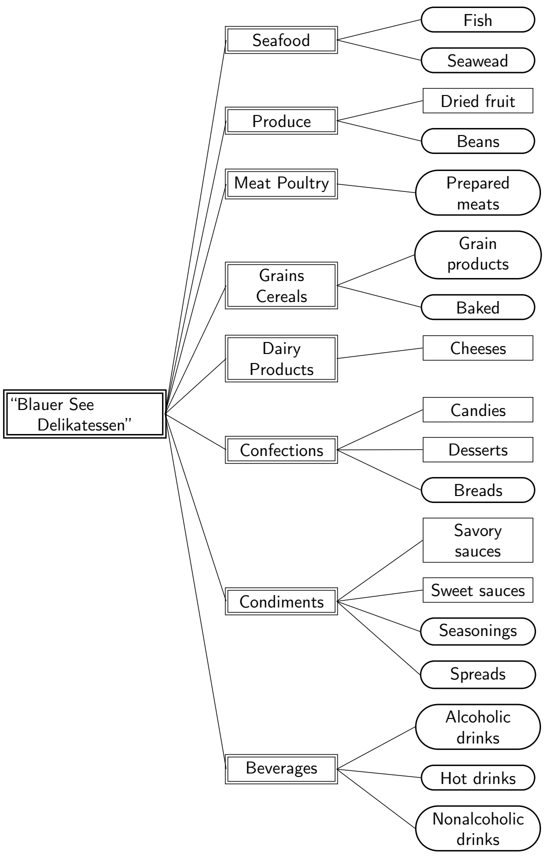


Fig. 9. The hierarchical structure of the personalized directory for the seafood market "Blauer See Delikatessen" ($CCI = 6$)

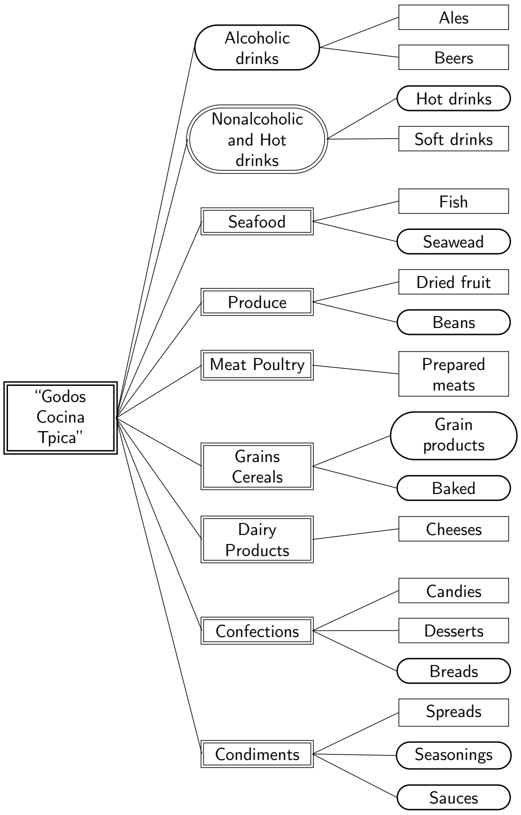


Fig. 10. The hierarchical structure of the personalized directory for the restaurant "Godos Cocina Topica" ($CCI = 30$).

5. Experimental evaluation

TaxoPublish is evaluated on two open databases (*AdventureWorks*[10], and *Northwind*), and one database provided by a retailing firm (*Festool*[11]). The characteristics of the used databases are summarized in Table 5. Hereby, each database is investigated against the tuples representing facts of the knowledge base. The facts *kcco*/3 and *kcpc*/2 are representing the companies/customers, as well as the persons being responsible for the companies' purchases. The facts *kcod*/2 and *kcoh*/2 express the details and headers for different orders. The facts *kppr*/3, *kpgr*/2, and *kpcl*/3 are

[10] http://msftdbprodsamples.codeplex.com/
[11] https://www.festool.de/

representing the e-catalog (PIM taxonomy) with its products, super concepts and sub concepts. The facts $kcse/2$ and $kcss/3$ are representing the taxonomy structuring the different branches of companies (CRM) with its super concepts and sub concepts, respectively. And, the facts $kodl/2$ and $kobl/2$ are necessary to generate the personalized directory with its dependencies, and bindings.

Table 5
Characteristics and parameters of the three different databases used for experimental results.

Predicate	AdventureWorks	Northwind	Festool
$kcco/3$	700	93	500
$kcpc/2$	700	93	608
$kcod/2$	121317	2155	1218
$kcoh/2$	31464	829	1400
$kppr/3$	320	77	118
$kpgr/2$	4	8	9
$kpcl/3$	37	22	43
$kcse/2$	4	5	21
$kcss/3$	15	18	59
$kodl/2$	14	16	21
$kobl/2$	52	6	32

Since the taxonomy overload problem for e-catalogs has not yet been addressed so far in the literature, the frameworks' single components are compared against different metrics:

- *Forecasting requirements* is evaluated with the help of the standard metrics used in information retrieval (Powers (2007)). The different metrics verify, how often a bundled dependency is performed and is actually out of interest in the recent epoch. *Precision* states how many bundled dependencies are irrelevant, with (see Equation 18):

$$Precision = \frac{\sum TP}{\sum TP + \sum FP}, \tag{18}$$

where TP is a true positive, and FP is a false positive statement. *Recall* states how many irrelevant dependencies are bundled, with (see Equation 19):

$$Recall = \frac{\sum TP}{\sum FN + \sum TP}, \tag{19}$$

where FN is a false negative statement. The *F-Measure* score, also referred as F_1 score, compares the harmonic mean of precision and recall to state the correctness of the system, with (see Equation 20):

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}. \tag{20}$$

And finally, *Accuracy* states how accurate the forecasting is on average, with (see Equation 21):

$$Accuracy = \frac{\sum TP + \sum TN}{\sum TP + \sum FP + \sum FN + \sum TN}. \tag{21}$$

- *Personalizing taxonomy* is evaluated to verify the decrease of the taxonomy overload. Hereby, we investigate the system against the provided problem formulation. To do so, each entire PIM taxonomy is firstly investigated against its initial number of sub concepts Φ_ψ, the minimum number of sub concepts Φ_{min} by reducing the sub concepts to its dependencies, and the number of actually required most detailing concepts Φ_ω for the different users. Consequently, the former and the latter result the number of irrelevant sub concepts Δ. Secondly, the number of concepts detailing a single super concept are investigated. Hereby, the number of initial sub concepts Φ_θ, and the number of sub concepts actually required Φ_θ are taken into account for resulting the number of distraction Ξ.
- *Computational efficiency* is investigated by providing the five standard runtime statistics in logic programming. The CPU in percent (Central Processing Unit) is used to indicate how much of the computers resources are required to treat the logic query. The CPU time is provided to state how long the CPU requires for the query. Both in common finally state the time in seconds required to execute the logic query. In addition, the number of LIPS (Logical Inferences Per Second), as well as the number of inferences required to satisfy the logic goal are given. In contrast to the CPU and time, which are metrics applied over all programming paradigms, this values highlight the logical performance efficiency of the system. The computational efficiency measure is divided into the two before-mentioned components: for forecasting requirements, and for personalizing the taxonomy.

5.1. Forecasting requirements

To provide the most comprehensive analysis for stating how accurate the forecasting of requirements is performing, the measures are divided into two major directions. This division provides two meaningful insights with respect to forecasting requirements for mediator concepts. On the one hand, it can state if the combination of a past-term analysis and future-term analysis can improve the forecasting. On the other hand, it can identify if the existing combination can be improved through domain-specific variables. To do so, firstly, the single components for the future-term analysis, in the following named Future-Term TaxoPublish, and for the past-term analysis, in the following named Past-Term TaxoPublish, are compared against when combining both components, referred as TaxoPublish. Secondly, the results of single components are compared to related works. Past-Term TaxoPublish is compared to the work presented by Ding and Li (2005), in the following shortened as Past-Term Ding. Past-Term Ding represents the epoch rate function used for the past term analysis, but without considering a variable decay rate. Future-Term TaxoPublish is compared with the work from Viswanathan and Krishnamurthi (2012), in the following shortened as Future-Term Viswanathan. Future-Term Viswanathan measures the equality between different customer concepts (groups), as used for the future-term analysis, but without a normalizing step to smooth the overload when comparing multiple similar users to a single customer.

The results provided in Figures 11, 12, and 13 show the improvement when combining the past-term analysis with a future-term analysis to effectively predict the upcoming interests. For all three databases, the combined recommendation technique TaxoPublish showed best results compared to Past-Term Ding and Future-Term Viswanathan. For the *Northwind* database, our system could improve the F-Measure score by 28.57% and by 7.14 %, respectively. For the *Adventureworks* database, an improve by 14.81 % and 17.72 % was performed. For the *Festool* database, the proposed expert system could improve the F-Measure score by 7.87 % and 2.13 %, respectively. However, this improvement is less than for the other two databases because of two reasons. Firstly, *Festool* is using the most comprehensive CRM taxonomy to classify the customers used for the future-term analysis (59 concepts). Through this, the fluctuations regarding upcoming preferences are lower than for the other databases (less than 20 concepts). Secondly, as the single customers inside the *Festool* database stay most loyal to their preferences, Past-Term TaxoPublish already shows a very good result for forecasting the requirements. However, TaxoPublish can still improve the forecasting for this database, as some companies are having multiple persons ordering for a single customer/company.

When adding a domain-specific epoch-rate value to Past-Term Ding, the forecasting can be furthermore increased for one of the investigated databases. More precisely, for the *Northwind*

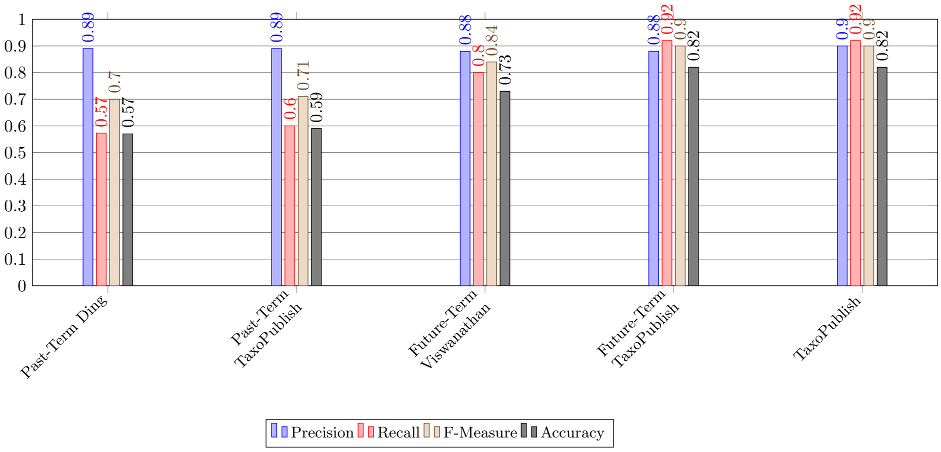


Fig. 11. *TaxoPublish* past-term (Past-Term TaxoPublish), future-term (Future-Term TaxoPublish), and combined (TaxoPublish) preference analysis results for the *Northwind* database, in comparison with existing works presented in Ding and Li (2005) (shortened as Past-Term Ding), and Viswanathan and Krishnamurthi (2012) (shortened as Future-Term Viswanathan).

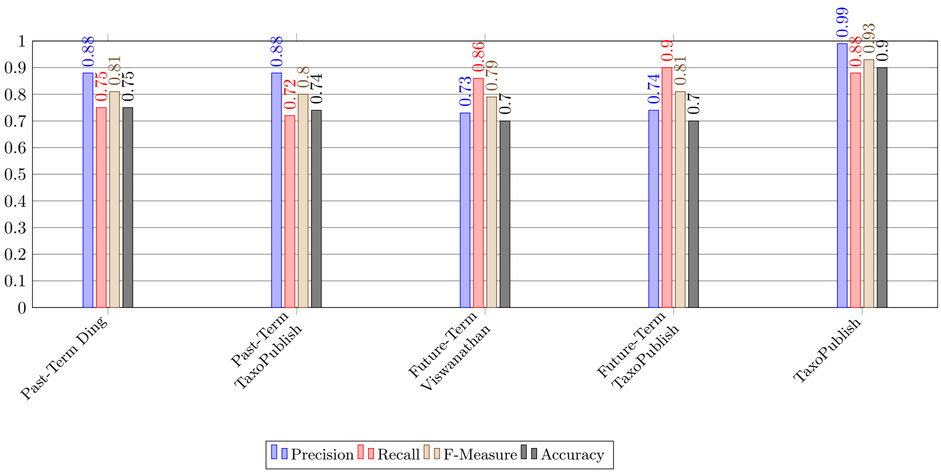


Fig. 12. *TaxoPublish* past-term (Past-Term TaxoPublish), future-term (Future-Term TaxoPublish), and combined (TaxoPublish) preference analysis results for the *Adventureworks* database, in comparison with existing works presented in Ding and Li (2005) (shortened as Past-Term Ding), and Viswanathan and Krishnamurthi (2012) (shortened as Future-Term Viswanathan).

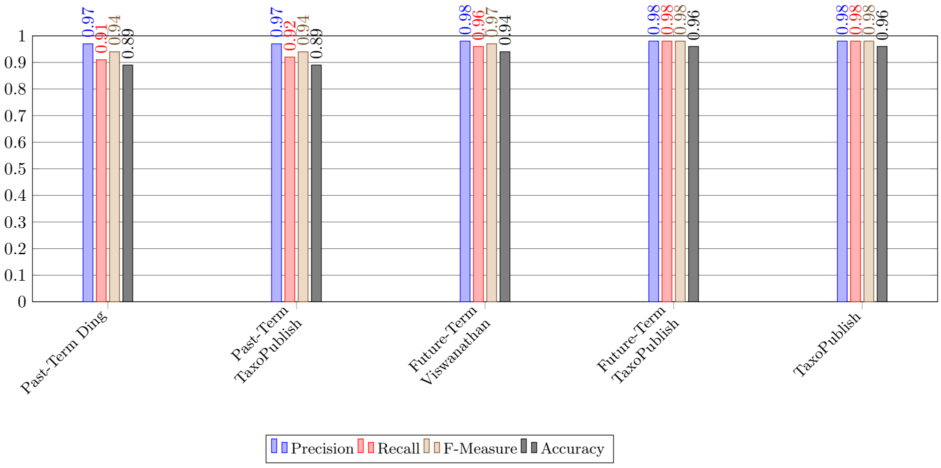


Fig. 13. *TaxoPublish* past-term (Past-Term TaxoPublish), future-term (Future-Term TaxoPublish), and combined (TaxoPublish) preference analysis results for the *Festool* database, in comparison with existing works presented in Ding and Li (2005) (shortened as Past-Term Ding), and Viswanathan and Krishnamurthi (2012) (shortened as Future-Term Viswanathan).

database by 1.43 %. This highlights that Past-Term TaxoPublish already performs with very high correctness over different e-commerce domains. Contrary, when adding a normalization step instead of only considering the semantic context weight as proposed in Future-Term Viswanathan, the forecasting for all three databases could be improved using Future-Term TaxoPublish. According to the F-Measure scores, an increase of 7.14 % was performed for the Northwind database. For the *Adventureworks* and *Festool* database, an increase by 2.53 % and 1.03 % was performed. This highlights that the included normalization step can overcome the lack of details inside the CRM taxonomy.

5.2. Personalizing taxonomy

To demonstrate the strength and weakness of the proposed system regarding personalizing the taxonomy, this process is compared against the provided problem formulation, against the modification rules presented in other four works dealing with personalized directories, as well as against other affected research areas: dynamic taxonomies, and catalog segmentation.

Table 6
Decreasing the taxonomy overload for non-preferred and high-preferred dependencies.

Variable	Northwind		AdventureWorks		Festool	
	$\bar{x}^{a}$	s^{b}	$\bar{x}^{a}$	s^{b}	$\bar{x}^{a}$	s^{b}
Φ_{ψ}	22	-	37	-	43	-
$\Phi_{m}in$	16	-	14	-	23	-
Φ_{ω}	20	2	24	6	24.68	3.64
Δ	2.46	1.56	12.70	6.31	18.32	3.64
Φ_{θ}	2.75	1.49	9.25	4.86	4.78	2.91
Φ_{ϑ}	1.38	0.52	2.47	1.06	1.80	0.83
Ξ	1.81	1.30	8.22	3.56	4.21	2.34

[a] $\bar{x}$ = Average Mean

[b] s = Standard Deviation

When summarizing the results corresponding to the taxonomic overload, a significant improvement can be achieved (see Table 6). Furthermore, it should be noted, that the reduced taxonomy overload depends on the size of the taxonomy along with its customers, and thus the efficiency of the approach increases with the number of sub concepts included in the taxonomy, see Figure 14. E.g., the *Northwind* database consists of only 22 sub concepts where on average 2.46 concepts can be reduced in the personalized taxonomy. Contrary, the *AdventureWorks* database consists of 37 sub concepts, where on average 12.70 concepts can be reduced when creating the personalized taxonomies. The efficiency of the moving operation also shows very proper results. For high preferred sub concepts, it is important to minimize the number of sibling concepts to emphasize the semantic context of the preferred sub concept, respectively its super concept. For all three databases, the number of sibling concepts is reduced for high preferred concepts. Again, the results depend on the size of the initial taxonomy. For the *Northwind* database, the number of siblings of highly preferred concepts is reduced by 1.81 concepts, whereas for the *AdventureWorks* database, highly preferred concepts had on average 9.25 siblings before the modification, but after the modification only 2.46 siblings.

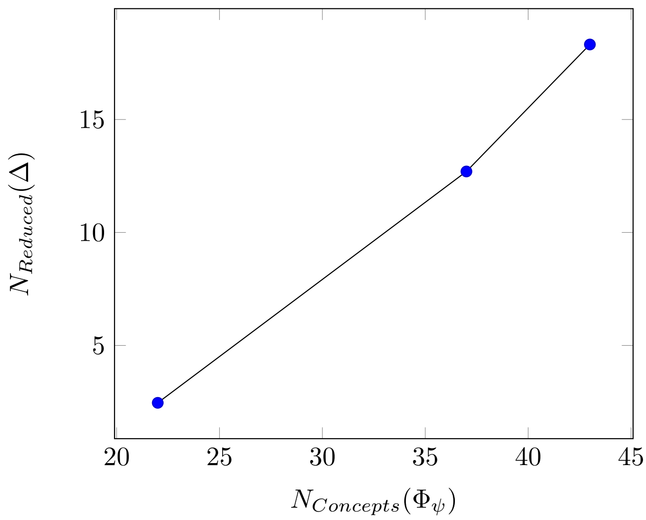


Fig. 14. Correlation between the reduction of the taxonomy overload and the size of the taxonomy.

5.2.1. Analytical Comparison with works on personalized directories

To highlight the strength and weakness of our propsed system, the modification rules presented in related works (personalized directories) are compared. Hereby, it is investigated, if the approach can be used to reduce the taxonomy overload, the technique is not missing information about the domain, and if the approach is performing fully-automatically.

As the modification rules presented in Joh and Lee (2003) are focussing on splitting sub concepts, splitting a subset of sub concepts to a higher level, splitting all sub concepts to a higher level, the shifting of a subset to a deeper level, and splitting sub concepts into more detailed sub concepts, their approach is mainly focussing on emphasizing preferred concepts. Only the sifting of a subset of sub concepts to a deeper level would reduce the taxonomy overload. However, this affects that a not required sibling concept still remains. In addition, it is contrary to the main idea of their approach, namely to minimize the depth of the taxonomy. The main drawback of this method is that the modifications have to be performed manually.

Contrary, the approach in Lin and Hong (2008) provides different database components for storing information about the customers, the products, and the taxonomies. Similar to our technique, they are utilizing a mining system to analyze customer preferences, which consists of six components: collecting customer data and transaction data, analyzing consumer behaviors, measuring segmentation and brand likings, for the knowledge acquisition for building the knowledge bases, and the catalog marketing and sales promotion component for the distribution of marketing strategies. However, the approach is not dividing the transaction data in weighted epochs, and thus can not react on dynamically changing user preferences. Additional, not preferred concepts will not be part of the new taxonomy to be created, and thus information about the domain will be lost.

The authors in Farsani and Nematbakhsh (2007) are providing an e-catalog management system to create, update, and customize individual taxonomies. It aims in helping the expert user (provider), as well as the customers. Their system requires feedback from the user in the form of keywords to analyze the customers' labeling preferences for making conceptual recommendations. Through this it works (semi)-automatically and could be used to reduce overload. However, it can not be expected that customers have a very high expertise about the domain. Thus, the creation of personalized taxonomies through the customer is not advisable.

The most recent work was presented in Fathy et al. (2014). An approach reordering search results by the help of concepts according to users interest. Their system is utilizing a concept-based user profile to learn customers preferences and to prune the taxonomy based on a given keyword. Consequently, it is using information about the domain, and is not performing fully-automatically.

5.2.2. Analytical Comparison with affected related areas: dynamic taxonomies, and catalog segmentation

To further demonstrate the impact of our proposed research method, we further compare the expert system against related research paradigms: dynamic taxonomies, and catalog segmentation. This works are very similar to the works in personalized directories and also deal with the reduction of the taxonomy overload. Again, it is investigated, if the paradigm can be used to reduce the taxonomy overload, if the techniques in the field are not missing information about the domain, and if the approaches are performing fully-automatically.

Dynamic taxonomies are based on a (static) taxonomy, which prunes itself in response to the request and so considers the significance of a user-query (Sacco et al. (2012)). These paradigm has been proposed as a solution to combine navigation and querying, offering both expressivity and interactivity (Ferre and Ridoux (2007)). Compared to personalized directories and catalog segmentation, this approaches offer the highest interactivity with the customer. The dynamic taxonomy can be changed according to changing user queries. However, the (semi)-automatic technique is missing information about the domain, as not detected matches against the provided keyword are not displayed inside the dynamic taxonomy.

Catalog segmentation is also based on a (static) taxonomy, but proposes to create a variety of sub-taxonomies for different segments of customers (Amiri (2006)). Through this, the considered customers can be removed or added to a specific customer group, which is assigned with a pre-defined and segmented catalog. This paradigm has been proposed as in many cases, some customers are interested in only a small segment of the goods the retailing firm offers (Amiri (2006)). This paradigm is most effectively reducing the taxonomy overload with explicitly excluding not preferred concepts. However, through this it significantly reduces information about the domain, and the loss can not be corrected through a new query as in the paradigm above.

5.2.3. Analytical Comparison: Strength and Weakness

When now comparing our method against the approaches in personalized directories, and against the related research areas, the strength and weakness of the proposed *TaxoPublish* expert system can be summarized, see Tables 8 and 7.

The most important strength of *TaxoPublish* is the reduction of the taxonomy without changing the semantics inside the taxonomy. Through this, the customer is also not distracted by non preferred concepts, but in the case of changing preferences, she/he is still able to reach all different products. Consequently and secondly, the provider of an e-catalog has no less chances to sell its products. For example, if a B2B customer is changing its business topic, *TaxoPublish* is capable to react on this changes through considering differently weighted epochs. Thirdly, as the final state of preference can be adapted dynamically and the modification is performing fully-automatically, *TaxoPublish* can also be used in nowadays important multi-channel e-commerce. For example, the bundling operation can be ignored for the digital imposed catalog, but can remain for the printed imposed catalog. Through this, the physical catalog has a limited number of pages, which safes resources and money on the side of the provider.

However, the reduction of the catalog can of course not be as much reduced as in catalog segmentation, the first drawback of *TaxoPublish*. Also if all concepts of a dependency are never desired over years, one single concept still remains. The second drawback of the proposed research method concerns the interaction with the customer. In dynamic taxonomies, the customer can interact with the taxonomy through providing a user-query, or through additional filters, so-called facets. *TaxoPublish* can not interact with the customer, as the usage of explicit knowledge would not allow a fully-automatic approach.

Table 7
Comparison of the presented system against related paradigms.

Paradigm	Overload Reduced	Semantic, Domain Remains	Fully-Automatic
Dynamic Taxonomies	✓	(✓)	
Catalog Segmentation	✓		✓
TaxoPublish	✓	✓	✓

Table 8
Comparison of the presented modification rules against related works.

Paradigm	Overload Reduced	Semantic, Domain Remains	Fully-Automatic
Joh and Lee (2003)	(✓)	✓	
Lin and Hong (2008)	✓		✓
Farsani and Nematbakhsh (2007)	✓	(✓)	(✓)
Fathy et al. (2014)	✓		(✓)
TaxoPublish	✓	✓	✓

5.3. Computational efficiency

For the computational efficiency measure the five most important runtime statistics values were taken into account: CPU in %, CPU time, time in seconds, LIPS (logical inferences per second), and inference. For obtaining the performance result, the system was developed using SWI Prolog, and the computational results were obtained on a 2.3 GHz Intel Core i5, 4 GB of RAM 1333 MHz DDR3 (see Tables 9 and 10). Through dividing the computational efficiency measure into the two components of the expert system (preference analysis, personalizing taxonomy) it can be investigated if there is a correlation between the sizes of the databases and the time required to perform the logic queries.

Table 9
Computational efficiency measure results to analyze customers preferences.

Variable	Northwind		AdventureWorks		Festool	
	$\bar{x}^{a}$	s^{b}	$\bar{x}^{a}$	s^{b}	$\bar{x}^{a}$	s^{b}
CPU in %[a]	92.60	0.52	92.60	0.01	91.80	1.03
CPU[b]	0.81	0.42	2.94	1.41	0.44	0.42
t in seconds	0.87	0.45	3.16	1.50	0.48	0.44
LIPS ($*10^3$)[d]	4533	3066	4353	1939	3783	7030
Inference ($*10^3$)[e]	3789	2181	1292	6377	1921	2246

[a] $\bar{x}$ = Average Mean

[b] s = Standard Deviation

From the performance results, it can be seen that the time to analyse the customers preference mainly depends on the number of epochs the customer has shown. For the *Adventureworks* database, the customers have on average 44.95 epochs, which results the longest time to analyze the preferences, namely on overage 3.16 seconds. In contrast, for the *Festool* database, the customers have on average 2.80 epochs, and the time required is on overage 0.48 seconds. Through this, there is a linear correlation between the time required to analyze the preferences, and the number of epochs a customer has shown. One possibility to solve this challenge, would be to remove the most distant epochs for the past-term analysis by taking the optimal trade-off into account. However, in a B2B scenario as well as in a B2C scenario, the preferred products often depend on seasonal circumstances. So, to only remove the epochs, as well as to only analyze the most recent epochs would significantly cause the accuracy of the preference analysis in a negative manner. The only possibility to remove epochs would be to improve the past-term analysis through providing a very detailed CRM taxonomy. However, this is only supported for the *Festool* database (59 concepts). The other two databases have less than 20 concepts to classify the customes, and through this, the prediction of preferences must be supported through a very accurate past-term analysis. Another aspect that has been considered, was to classify the epochs in a seasonal manner, e.g. to provide four different epoch groups. However, in a real-time expert system, the classification into epoch groups did not show an improvement, as this requires a furthermore logic query to be performed.

Table 10
Computational efficiency measure results to fully-automatically create personalized taxonomies.

Variable	Northwind		AdventureWorks		Festool	
	$\bar{x}^{a}$	s^{b}	$\bar{x}^{a}$	s^{b}	$\bar{x}^{a}$	s^{b}
CPU in %	40.60	26.02	31.00	25.21	30.40	21.50
CPU	0.01	0.00	0.01	0.00	0.01	0.00
t in seconds	0.02	0.02	0.03	0.03	0.03	0.02
LIPS ($*10^3$)	1384	123	1589	3181	1705	3257
Inference	7090	322	7262	3118	8334	5148

[a] $\bar{x}$ = Average Mean

[b] s = Standard Deviation

Another linear correlation exists between the time required to personalize the taxonomies, and the number of concepts used to describe the actually required taxonomy. The time for each database can be improved when the dependencies would be assigned to a lower number of sub concepts. However, than the time for the preference analysis would increase, and the taxonomy overload reduction would decrease by the relative number of additional dependencies.

6. Conclusion

This work presented *TaxoPublish*, the first solution for personalizing e-catalogs in a full-automatic manner by making use of an integrated recommender system implemented in logic programming.

Through the personalization of the e-catalog it remedies three major drawbacks the formal e-catalog suffers from. Firstly and foremost, the problem of the taxonomic overload is reduced. It does this by providing different modification rules depending on the customers preferences. Low preferred concepts can be combined, and high preferred concepts can be semantically enriched. The domain-specific requirements for detailing the taxonomy are overcome through providing scalable rules depending on the scope of the e-catalog marketplace. Secondly, the inflexibility of the taxonomy is reduced without missing any information about the domain. Hereby, it makes use of two novel taxonomic terms, namely taxonomy dependencies and taxonomy bindings. The dependencies are used as flexible mediator between a super and a subset of sibling sub concepts.

The bindings are used to react on the semantic changes, when modifications are performed for the dependencies. Thirdly, human effort is significantly reduced as the modification is performed in a fully-automatic manner. Through the usage of an included recommender system combining past-term and future-term analysis, the customers preferences can be effectively predicted. The developed system is implemented in logic programming and uses knowledge as provided in all leading e-commerce applications. Through this, the expert system is applicable to all in e-commerce used database paradigms and data warehause scenarios, hierarchical as well as relational databases and distributed architectures. The extensive case study has highlighted that the proposed system improves the usability of the taxonomy significantly. An evaluation performed on three databases highlights the capability of *TaxoPublish* to personalize the e-catalog without any human effort, but with an accuracy similar to the expert user.

Future work on *TaxoPublish* can be divided in four directions: the usage of background knowledge, the modification process for groups, the analyzis of product reviews, and a compressed storage of the resulting taxonomies. Background knowledge is recently used by taxonomy matching approaches to infer similarity between concepts, respectively taxonomies. Through the linking of *TaxoPublish* with background knowledge, e.g. in the semantic lexicon WordNet, it will be able to dynamically create the mediator concepts. The modification process is recently considered for "B2B" retailing markets. However, "B2B" retailing markets often also propose group-specific taxonomies (e.g. *Festool*), which can be achieved in *TaxoPublish* by reducing the integrated recommender systems. Additional work on the modification rules will be the extension of the proposed bundling operation. Until now, the bundling operation is combining the sub concepts. The combination can be enhanced through still combining the sub concepts, but with moving the single sub concepts to a deeper level inside the taxonomy. And, the moving operation is until now only considered for dependencies. A modification rule to also move single sub concepts to a higher level would be useful for very high preferred single sub concepts. However, this both suggested further modification rules would only improve the personalization, but would decrease the reduction of the taxonomy overload. Finally, a compression method will be included for storing the personalized taxonomies. As nowadays digital marketplaces usually have a very large number of customers compared to the number of possible taxonomies, this will further increase the performance.

Acknowledgement

Thanks to the Festool GmbH & Co. KG for providing data to analyse in our experiments. We are grateful to the reviewers for their constructive and very helpful input.

References

Agichtein, E., Brill, E., Dumais, S., Ragno, R., 2006. Learning user interaction models for predicting web search result preferences, in: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Association for Computing Machinery (ACM), New York, NY, USA. pp. 3–10. URL: http://doi.acm.org/10.1145/1148170.1148175, doi:10.1145/1148170.1148175.

Amiri, A., 2006. Customer-oriented catalog segmentation: Effective solution approaches. Decis. Support Syst. 42, 1860–1871. URL: http://dx.doi.org/10.1016/j.dss.2006.04.010, doi:10.1016/j.dss.2006.04.010.

Basu Roy, S., Wang, H., Das, G., Nambiar, U., Mohania, M., 2008. Minimum-effort driven dynamic faceted search in structured databases, in: Proceedings of the 17th ACM Conference on Information and Knowledge Management, Association for Computing Machinery (ACM), New York, NY, USA. pp. 13–22. URL: http://doi.acm.org/10.1145/1458082.1458088, doi:10.1145/1458082.1458088.

Bonino, D., Corno, F., Farinetti, L., 2009. Faset: A set theory model for faceted search, in: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 01, IEEE Computer Society, Washington, DC, USA. pp. 474–481. URL: http://dx.doi.org/10.1109/WI-IAT.2009.81, doi:10.1109/WI-IAT.2009.81.

Bramer, M., 2014. Logic Programming with Prolog. 2nd ed., Springer Publishing Company, Incorporated.

Calegari, S., Pasi, G., 2013. Personal ontologies: Generation of user profiles based on the yago ontology. Inf. Process. Manage. 49, 640–658. URL: http://dx.doi.org/10.1016/j.ipm.2012.07.010, doi:10.1016/j.ipm.2012.07.010.

Conen, W., Klapsing, R., 2001. Exchanging semantics with rdf, in: Information Age Economy. Springer, pp. 473–486.

Ding, Y., Li, X., 2005. Time weight collaborative filtering, in: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, Association for Computing Machinery (ACM), New York, NY, USA. pp. 485–492. URL: http://doi.acm.org/10.1145/1099554.1099689, doi:10.1145/1099554.1099689.

Donglin, C., Xiaofei, L., Jun, Z., 2010. User-oriented intelligent service of e-catalog based on semantic web, in: The 2nd IEEE International Conference on Information Management and Engineering, Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA. pp. 449–453. doi:10.1109/ICIME.2010.5477872.

Döring, S., Kießling, W., Preisinger, T., Fischer, S., 2006. Evaluation and optimization of the catalog search process of e-procurement platforms. Electronic Commerce Research and Applications 5, 44–56. doi:10.1016/j.elerap.2005.08.004.

Farsani, H.K., Nematbakhsh, A., 2007. Designing a catalog management system– an ontology approach. Malaysian Journal of Computer Science 20, 119–127. doi:10.1.1.462.6537.

Fathy, N., Badr, N., Gharib, T., 2014. Personalized web search ranking based on different information resources, in: Das, V., Elkafrawy, P. (Eds.), Signal Processing and Information Technology. Springer International Publishing. volume 117 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 211–214. URL: http://dx.doi.org/10.1007/978-3-319-11629-7_31, doi:10.1007/978-3-319-11629-7_31.

Ferre, S., Ridoux, O., 2007. Logical information systems: From taxonomies to logics, in: Proceedings of the 18th International Conference on Database and Expert Systems Applications, IEEE Computer Society, Washington, DC, USA. pp. 212–216. URL: http://dx.doi.org/10.1109/DEXA.2007.97, doi:10.1109/DEXA.2007.97.

George, M., Kumar, V., Grewal, D., 2013. Maximizing profits for a multi-category catalog retailer. Journal of Retailing 89, 374 – 396. URL: http://www.sciencedirect.com/science/article/pii/S0022435913000390, doi:http://dx.doi.org/10.1016/j.jretai.2013.05.001.

Gollub, T., Völske, M., Hagen, M., Stein, B., 2014. Dynamic taxonomy composition via keyqueries, in: Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries, IEEE Press, Piscataway, NJ, USA. pp. 39–48. URL: http://dl.acm.org/citation.cfm?id=2740769.2740777.

Gomez-Perez, A., Fernández-López, M., Corcho, O., 2006. Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. Springer Science & Business Media.

Guarino, N., Oberle, D., Staab, S., 2009. What is an ontology?, in: Staab, S., Studer, R. (Eds.), Handbook on Ontologies. Springer-Verlag GmbH, Berlin/Heidelberg, Germany. volume 2 of *International Handbooks on Information Systems*, pp. 1–17. URL: http://dx.doi.org/10.1007/978-3-540-92673-3_0, doi:10.1007/978-3-540-92673-3_0.

Joh, Y.H., Lee, J.K., 2003. Buyer's customized directory management over seller's e-catalogs: Logic programming approach. Decis. Support Syst. 34, 197–212. URL: http://dx.doi.org/10.1016/S0167-9236(02)00081-7, doi:10.1016/S0167-9236(02)00081-7.

Kumaraguru, S., Rachuri, S., Lechevalier, D., 2014. Faceted classification of manufacturing processes for sustainability performance evaluation. The International Journal of Advanced Manufacturing Technology 75, 1309–1320. URL: http://dx.doi.org/10.1007/s00170-014-6184-x, doi:10.1007/s00170-014-6184-x.

Li, L., Hong, W., Li, T., 2012. Taxonomy-oriented recommendation towards recommendation with stage, in: Proceedings of the 14th Asia-Pacific International Conference on Web Technologies and Applications, Springer-Verlag GmbH, Berlin/Heidelberg, Germany. pp. 219–230. URL: http://dx.doi.org/10.1007/978-3-642-29253-8_19, doi:10.1007/978-3-642-29253-8_19.

Liang, H., Xu, Y., Li, Y., Nayak, R., Tao, X., 2010. Connecting users and items with weighted tags for personalized item recommendations, in: Proceedings of the 21st ACM Conference on Hypertext and Hypermedia, Association for Computing Machinery (ACM), New York, NY, USA. pp. 51–60. URL: http://doi.acm.org/10.1145/1810617.1810628, doi:10.1145/1810617.1810628.

Liao, S.H., Chen, C.M., Wu, C.H., 2008. Mining customer knowledge for product line and brand extension in retailing. Expert Syst. Appl. 34, 1763–1776. URL: http://dx.doi.org/10.1016/j.eswa.2007.01.036, doi:10.1016/j.eswa.2007.01.036.

Lin, C., Hong, C., 2008. Using customer knowledge in designing electronic catalog. Expert Systems with Applications 34, 119–127. doi:doi:10.1016/j.eswa.2006.08.028.

Mahdavi, I., Movahednejad, M., Adbesh, F., 2011. Designing customer-oriented catalogs in e-crm using an effective self-adaptive genetic algorithm. Expert Systems with Applications 38, 631 – 639. URL: http://www.sciencedirect.com/science/article/pii/S0957417410006305, doi:http://dx.doi.org/10.1016/j.eswa.2010.07.013.

Merrit, D., 2000. Expert systems with Prolog. Springer-Verlag, Berlin.

Ostermayer, L., Seipel, D., 2012. Knowledge engineering for business rules in prolog, in: Proc. Workshop on Logic Programming (WLP), pp. 1–24.

Pazienza, M., Pennacchiotti, M., Zanzotto, F., 2005. Terminology extraction: An analysis of linguistic and statistical approaches, in: Sirmakessis, S. (Ed.), Knowledge Mining. Springer-Verlag GmbH, Berlin/Heidelberg, Germany. volume 185 of *Studies in Fuzziness and Soft Computing*, pp. 255–279. URL: http://dx.doi.org/10.1007/3-540-32394-5_20, doi:10.1007/3-540-32394-5_20.

Phan, D.D., Vogel, D.R., 2010. A model of customer relationship management and business intelligence systems for catalogue and online retailers. Inf. Manage. 47, 69–77. URL: http://dx.doi.org/10.1016/j.im.2009.09.001, doi:10.1016/j.im.2009.09.001.

Pierrakos, D., Paliouras, G., 2010. Personalizing web directories with the aid of web usage data. Knowledge and Data Engineering, IEEE Transactions on 22, 1331–1344. doi:10.1109/TKDE.2009.173.

Powers, D.M.W., 2007. Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. Technical Report. Flinders University.

Raunich, S., Rahm, E., 2012. Towards a benchmark for ontology merging, in: Herrero, P., Panetto, H., Meersman, R., Dillon, T. (Eds.), On the Move to Meaningful Internet Systems: OTM 2012 Workshops. Springer-Verlag GmbH, Berlin/Heidelberg, Germany. volume 7567 of *Lecture Notes in Computer Science*, pp. 124–133. URL: http://dx.doi.org/10.1007/978-3-642-33618-8_20, doi:10.1007/978-3-642-33618-8_20.

Sabater-Mir, J., Cuadros, J., Garcia, P., 2013. Towards a framework that allows using a cognitive architecture to personalize recommendations in e-commerce., in: EUMAS, pp. 3–17.

Sacco, G.M., Nigrelli, G., Bosio, A., Chiarle, M., Luino, F., 2012. Dynamic taxonomies applied to a web-based relational database for geo-hydrological risk mitigation. Comput. Geosci. 39, 182–187. URL: http://dx.doi.org/10.1016/j.cageo.2011.07.005, doi:10.1016/j.cageo.2011.07.005.

Schulten, E., Akkermans, H., Botquin, G., Dörr, M., Guarino, N., Lopes, N., Sadeh, N., 2001. Call for participants: The e-commerce product classification challenge. IEEE Intelligent Systems 16, 86–c3. URL: http://dl.acm.org/citation.cfm?id=630319.630663, doi:n.a.(ACMID:630663).

Shen, X., Tan, B., Zhai, C., 2005. Context-sensitive information retrieval using implicit feedback, in: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Association for Computing Machinery (ACM), New York, NY, USA. pp. 43–50. URL: http://doi.acm.org/10.1145/1076034.1076045, doi:10.1145/1076034.1076045.

Tao, X., Li, Y., Zhong, N., 2011. A personalized ontology model for web information gathering. Knowledge and Data Engineering, IEEE Transactions on 23, 496–511. doi:10.1109/TKDE.2010.145.

Teevan, J., Dumais, S.T., Horvitz, E., 2010. Potential for personalization. ACM Trans. Comput.-Hum. Interact. 17, 4:1–4:31. URL: http://doi.acm.org/10.1145/1721831.1721835, doi:10.1145/1721831.1721835.

Álvaro Tejeda-Lorente, BernabÈ-Moreno, J., Porcel, C., Galindo-Moreno, P., Herrera-Viedma, E., 2015. A dynamic recommender system as reinforcement for personalized education by a fuzzly linguistic web system. Procedia Computer Science 55, 1143 – 1150. URL: http://www.sciencedirect.com/science/article/pii/S1877050915015598, doi:http://dx.doi.org/10.1016/j.procs.2015.07.084. 3rd International Conference on Information Technology and Quantitative Management, {ITQM} 2015.

Tvarozek, M., Bielikova, M., 2007. Personalized faceted navigation for multimedia collections, in: Proceedings of the 2nd International Workshop on Semantic Media Adaptation and Personalization, IEEE Computer Society, Washington, DC, USA. pp. 104–109. URL: http://dx.doi.org/10.1109/SMAP.2007.33, doi:10.1109/SMAP.2007.33.

Vandic, D., Van Dam, J.W., Frasincar, F., 2012. Faceted product search powered by the semantic web. Decis. Support Syst. 53, 425–437. URL: http://dx.doi.org/10.1016/j.dss.2012.02.010, doi:10.1016/j.dss.2012.02.010.

Viswanathan, V., Krishnamurthi, I., 2012. Ranking semantic associations between two entities — extended model, in: Proceedings of the 4th Asian Conference on Intelligent Information and Database Systems - Volume Part III, Springer-Verlag GmbH, Berlin/Heidelberg, Germany. pp. 152–162. URL: http://dx.doi.org/10.1007/978-3-642-28493-9_17, doi:10.1007/978-3-642-28493-9_17.

Xu, X., Liu, Y., Wang, Z., Zhou, C., Liang, Y., 2009. Catalog segmentation with double constraints in business. Pattern Recogn. Lett. 30, 440–448. URL: http://dx.doi.org/10.1016/j.patrec.2008.10.014, doi:10.1016/j.patrec.2008.10.014.

Xu, Z., Du, D., Xu, D., 2014. Improved approximation algorithms for the max-bisection and the disjoint 2-catalog segmentation problems. Journal of Combinatorial Optimization 27, 315–327. URL: http://dx.doi.org/10.1007/s10878-012-9526-3, doi:10.1007/s10878-012-9526-3.