Pricing Information Goods in Distributed Agent-Based Information Filtering*

Christos Tryfonopoulos[1] and Laura Maria Andreescu[2]

[1] University of Peloponnese, Tripoli, Greece
trifon@uop.gr
[2] APYDOS, Luxembourg
landreescu@apydos.com

Abstract. Most approaches to information filtering taken so far have the underlying hypothesis of potentially delivering notifications from every information producer to subscribers; this exact information filtering model creates efficiency and scalability bottlenecks and incurs a cognitive overload to the user. In this work we put forward a distributed agent-based information filtering approach that avoids information overload and scalability bottlenecks by relying on approximate information filtering. In approximate information filtering, the user subscribes to and monitors only carefully selected data sources, to receive interesting events from these sources only. In this way, system scalability is enhanced by trading recall for lower message traffic, information overload is avoided, and information producers are free to specialise, build their subscriber base and charge for the delivered content. We define the specifics of such an agent-based architecture for approximate information filtering, and introduce a novel agent selection mechanism based on the combination of resource selection, predicted publishing behaviour, and information cost to improve publisher selection. To the best of our knowledge, this is the first approach to model the cost of information in a filtering setting, and study its effect on retrieval efficiency and effectiveness.

1 Introduction

Much information of interest to humans is available today on the Web, making it extremely difficult to stay informed without sifting through enormous amounts of information. In such a dynamic setting, *information filtering (IF)*, also referred to as *publish/subscribe*, *continuous querying*, or *information push*, is equally important to one-time querying, since users are able to subscribe to information sources and be notified when documents of interest are published. This need for *content-based* push technologies is also stressed by the deployment of new tools such as Google Alert. In an IF scenario, a user posts a *subscription* (or *continuous query*) to the system to receive *notifications* whenever certain events of interest take place (e.g., when a document on Special Olympics becomes available). Since in an IF scenario the data is originally highly distributed residing on millions of sites (e.g., with people contributing to blogs,

* Part of the work was done while the authors were with Max-Planck Institute for Informatics. The authors would like to thank David Midgley for his comments and suggestions on the economic aspects of this work.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011, Part I, LNCS 7044, pp. 163–181, 2011.

news portals, social networking feeds), a distributed approach seems an ideal candidate for such a setting.

In this work we put forward ABIS (*Agent-Based Information filtering System*), a novel agent-based architecture that supports content-based *approximate information filtering*. While most *exact information filtering* approaches [32,15,14,34,1,27,8] taken so far have the underlying hypothesis of potentially delivering notifications from every information producer, ABIS relaxes this assumption by monitoring only selected sources that are likely to publish documents relevant to the user's interests in the future. In ABIS, a user subscribes with a continuous query and monitors only the most interesting sources in the system, to receive published documents from these sources only. The system is responsible for managing the user query, discovering new potential sources and moving queries to better or more promising publishers. Approximate IF improves the scalability issues of exact IF by trading recall for lower message traffic, avoids information overload to the user, by allowing him to receive selected notifications from selected publishers, and proves an interesting business model for pricing information goods delivered by information producers. In approximate IF, each information producer might have its own customer base of interested subscribers, and may charge the delivered content by subscription or per item. Notice that this is not possible in the case of exact IF, since information consumers receive *all* matching notifications, from all producers, while to facilitate the distribution of the service, no notion of ownership control and publisher quality is employed. Finally, notice that system throughput and notification latency in exact IF depend heavily on publication size (which is usually large for textual IF). On the other hand, approximate IF is not affected by publication size (as there is no notion of information dissemination at publication time) and offers one-hop latency, since each publisher maintains its own database of subscribers. The interested reader is referred to [36] for an insightful comparison of exact and approximate IF.

As possible application scenarios for ABIS consider the case of news filtering (but with the emphasis on information quality rather than timeliness of delivery) or blog filtering where users subscribe to new posts. Not only do these settings pose scalability challenges, but they would also incur an information avalanche and thus cognitive overload to the subscribed users, if the users were alerted for each and every new document published at any source whenever this matched a submitted continuous query. Our approximate IF approach ranks sources, and delivers matches only from the best ones, by utilising novel publisher selection strategies. These strategies take into consideration the quality of the information publisher, based on per-publisher statistics, and the price of information as this is set by the publisher. Despite the utilisation of a *Distributed Hash Table* (DHT) [31] to maintain publisher statistics, notice that our architecture can also be realised in other settings, like a single coordinator agent monitoring a number of distributed sources, or a cloud-based multi-agent system providing an alerting service.

To the best of our knowledge, this is the first approach that aims at connecting system efficiency and effectiveness with the cost component, and puts economic modelling in the picture of distributed IF. In the light of the above, the contributions presented in this work are threefold:

- We define an agent-based architecture and its related protocols to support *approximate* IF functionality in a distributed multi-agent environment. This is the first approach to model approximate IF in an agent-based setting.
- We show that traditional resource selection strategies are not sufficient in approximate IF, and devise a *novel* method to rank publishers according to their expertise, their predicted publishing behavior (based on time-series analysis of IR metrics) and the price of the information goods they publish. This method allows us to achieve high recall, while monitoring only a small number of publishers.
- We study the effect of introducing a price component in an IF setting and experimentally demonstrate that price of information is a key element, that may have an significant effect on recall observed by the subscribers. Our modelling utilises concepts such as correlation between the quality/expertise of the publisher and the price it charges for information goods, computation of this price depending on the demand, and charging agents for utilisation of resources such as local agent and network utilisation.

In previous work, we have compared exact and approximate information filtering in [36,44], applied approximate IR and IF to the digital library domain [46], and investigated different time series analysis methods [45]. The current paper extends the core ideas behind approximate IF in a multi-agent architecture, and emphasises the price component and its effect on system effectiveness and efficiency.

The rest of the paper is organised as follows. Related work is discussed in Section 2. Section 3 presents the ABIS architecture, implemented services and agent protocols, while Section 4 introduces our agent selection method. Experimental results are presented in Section 5, and Section 6 concludes this paper.

2 Related Work

In this section we discuss related work in the context of pricing information goods in agent-based systems, and IF in distributed (e.g., multi-agent, P2P) environments.

2.1 Pricing of Information in Agent-Based Models

Information has the property of non-rivalrous consumption, contrary to other goods such as cars and apples that need to be produced individually in order to be consumed individually, and once purchased are removed from the market for subsequent buyers.

One distinct feature of information goods is that they have large fixed costs of production, and small variable costs of reproduction, which makes value-based more appropriate than cost-based pricing [40]. Different consumers may have radically different values for the same information good, so techniques for differential pricing become very important. The best known form of differential pricing is called quality discrimination or versioning [39]. Using versioning, the producer will divide the consumers into different groups according to their willingness to pay, or choose the price of the versions and their compelling features to induce the consumers to “self select" into appropriate categories [40].

Different pricing strategies are relevant for different disciplines and applications. It is important to be aware of the fact that not all services will necessarily be provided to all users. In [19] a thorough analysis on database pricing strategies is carried out, and different strategies (such as connect-time pricing, per-record charge, and value-based pricing) are identified and compared. In [21] complex adaptive systems are used to analyse pricing decisions in an industry with products that can be pirated, while [7] looks into pricing models for information from Bloomberg, Reuters and Bridge, and presents the advantages of subscription, two-tier pricing schemes (flat fee for subscription and then a small charge for every item ordered) and n-tier pricing schemes.

Available research also offers a variety of models that can be used to study the cost of transactions between agents in a market model. [18] models the cost of a product on two dimensions: a price for the product itself and a transaction cost. The transaction cost also has two components: a component that is correlated to the amount of data that needs to be transported through the network and a component that is based upon changes in quantities ordered. Along the same line, [38,41] discuss incentives for different pricing schemes for information goods, distinguishing between pure competitive markets (several producers of an identical commodity) and markets where producers have market power. In [13] the problem of using services provided from other agents is considered, while [11] presents a case-study on file-transfer and focuses on the utilisation factor of the links between agents.

2.2 Distributed Information Filtering

Research on distributed processing of continuous queries has its origins in SIENA [4], and extensions on the core ideas of SIENA, such as DIAS [22] and P2P-DIET [23,17].

With the advent of DHTs such as CAN, Chord and Pastry, a new wave of publish/subcribe systems has appeared. Scribe [30], Hermes [27], HYPER [42], Meghdoot [15], PeerCQ [14], and many others [36,1,44,8,34] utilised a DHT to build a content-based system for processing continuous queries.

Many systems also employed an IR-based query language to support information filtering on top of structured overlay networks have been deployed. DHTrie [34], Ferry [43], and [2], extended the Chord protocol [31] to achieve exact information filtering functionality and applied document-granularity dissemination to achieve the recall of a centralised system. In the same spirit, LibraRing [33] presented a framework to provide information retrieval and filtering services in two-tier digital library environments. Similarly, pFilter [32] used a hierarchical extension of the CAN DHT [29] to store user queries and relied on multi-cast trees to notify subscribers. In [1], the authors show how to implement a DHT-agnostic solution to support prefix and suffix operations over string attributes in a publish/subcribe environment.

Information filtering and retrieval have also been explored in the context of multi-agent systems. In [28] the design of a distributed multi-agent information filtering system called D-SIFTER is presented, and the effect of inter-agent collaboration on filtering performance is examined. In [24], a peer-to-peer architecture for agent-based information filtering and dissemination, along with the associated data models and languages for appropriately expressing documents and user profiles is presented. Finally,

the MAWS system [16] utilises mobile agents to reduce the volume of irrelevant links returned by typical search engines.

Query placement, as implemented in exact information filtering approaches such as [1,32,34], is deterministic, and depends upon the terms contained in the query and the hash function provided by the DHT. These query placement protocols lead to filtering effectiveness of a centralised system. Compared to a centralised approach, [1,32,34] exhibit scalability, fault-tolerance, and load balancing at the expense of high message traffic at publication time. In ABIS however, only the most promising agents store a user query and are thus monitored. Publications are only matched against its local query database, since, for scalability reasons, no publication forwarding is used. Thus, in the case of approximate filtering, the recall achieved is lower than that of exact filtering, but document-granularity dissemination to the network is avoided.

3 Services and Protocols in ABIS

In this section we present the services implemented in ABIS and the respective protocols that regulate agent interactions.

3.1 Types of Services

Within the multi-agent system we can distinguish between three types of services: a directory service, a publication service and subscription service. All agents in ABIS implement the directory service, and one or both of the publication and subscription service, depending whether they want to act as information producers, consumers or both.

Directory Service. The directory service manages aggregated statistical meta-information about the documents that are offered by publishers (i.e., aggregated statistical information for terms, prices per document). The role of this service is to serve as a global meta-data index about the documents and the prices available on the market. This index is partitioned among all agents in ABIS and is utilised by the subscribers to determine which publishers are promising candidates to satisfy a given continuous query in the future. There are different alternatives to implementing this type of directory, ranging from centralised solutions that emphasise accuracy in statistics and rely on server farms, to two-tier architectures. In our approach, we utilise a distributed directory of agents organised under a Chord DHT [31] to form a conceptually global, but physically distributed directory. The directory manages the statistics provided by the publishers in a scalable manner with good properties regarding system dynamics (e.g., churn). The DHT is used to partition the term space, such that every agent is responsible for the statistics of a randomised subset of terms within the directory. Hence, there is a well defined directory agent responsible for each term (through the DHT hash function).

Publication Service. The publication service is implemented by information producers (e.g., digital libraries or agents with local crawlers that perform focused crawling at portals of their interest) or users that are interested in selling their content. The publishers do not a priori know how much a subscriber is willing to pay for information

from his domain. In an ideal model the publishers will adjust the price according to the demand from the market: when a publisher is overloaded with requests, he would increase the price for the information he is offering. An agent implementing only the publication service creates meta-data for the resources it stores and uses the directory service to offer them to the rest of the network.

Each publisher exposes its content to the directory in the form of per-term statistics about its local index. These posts contain contact information about publishers, together with statistics to calculate quality measures and prices for a given term (e.g., frequency of occurrence). Typically, such statistics include quality measures to support the publisher ranking procedure carried out by subscribers, and are updated after a certain number of publications occurs. Finally, publishers are responsible for locally storing continuous queries submitted by subscribers and matching them against new documents they publish.

Subscription Service. The agents implementing the subscription service are information consumers, which subscribe to publications and receive notifications about resources that match their interests. The goal of the subscribers is to satisfy their long-term information needs by subscribing to publishers that will publish interesting documents in the future. A subscriber has access to all prices set by publishers for certain resource types through the directory service, and the subscribers are free to choose the best offer from the market that suits their needs and budget. To do so, subscribers utilise directory statistics to score and rank publishers, based on appropriate publisher selection and behaviour prediction strategies, as well as on the actual price of the requested item and the budget of the agent as we will discuss in following sections. To follow the changes in the publishing behaviour of information producers, subscribers periodically re-rank publishers by obtaining updated statistics from the directory, and use the new publisher ranking to reposition their continuous queries.

3.2 The ABIS Protocols

All agents implementing the aforementioned services follow a specific protocol to facilitate message exchange in a scalable way. Below we describe the protocols that facilitate agent interaction, for each one of the described services.

The Directory Protocol. The directory service manages aggregated information about each agent's local knowledge in a compact form. Every agent is responsible for the statistics for a randomised subset of terms within the directory. To keep the statistics up-to-date, each agent distributes per-term summaries of its local index along with its contact information. For efficiency reasons, these messages are piggy-backed to DHT maintenance messages and batching is used.

To facilitate message sending between agents we will use the function SEND(msg, I) to send message *msg* to the agent responsible for identifier I. Function SEND() is similar to the Chord function LOOKUP(I) [31], and costs $O(logN)$ overlay hops for a network of N agents. In ABIS, every publisher uses POST messages to distribute per-term statistics. This information is periodically updated (e.g., every k time units or every k publications) by the publisher agent, in order to keep the directory information as up-to-date

as possible. Let us now examine how a publisher agent P updates the global directory. Let $T = \{t_1, t_2, \ldots, t_k\}$ denote the set of all terms contained in all document publications of P occurring after the last directory update. For each term t_i, where $1 \leq i \leq k$, P computes the maximum frequency of occurrence of term t_i within the documents contained in P's collection ($tf_{t_i}^{max}$), the number of documents in the document collection of P that t_i is contained in (df_{t_i}), and the size of the document collection cs. Having collected the statistics for term t_i, P creates message POST$(id(P), ip(P), tf_{t_i}^{max}, df_{t_i}, cs, t_i)$, where $id(P)$ is the identifier of agent P and $ip(P)$ is the IP address of P. P then uses function SEND() to forward the message to the agent responsible for identifier $H(t_i)$ (i.e., the agent responsible for maintaining statistics for term t_i). Once an agent D receives a POST message, it stores the statistics for P in its local post database to keep them available on request for any agent.

Finally, notice that the directory service does not have to use Chord, or any other DHT; our architecture allows for the usage of any network structure given that the necessary information (i.e., the per-agent IR statistics) is made available through appropriate protocols to the rest of the services.

The Subscription Protocol. The subscription service is implemented by agents that want to monitor specific information producers. This service is critical since it is responsible for selecting the publishers that will index a query. This procedure uses the directory service to discover and retrieve the publishers that have information on a given topic. Then a ranking of the potential sources is performed and the query is send to top-k ranked publishers. Only these publishers will be monitored for new publications.

Let us assume that a subscriber agent S wants to subscribe with a *multi-term query* q of the form $t_1 t_2 \ldots t_k$ with k distinct terms. To do so, S needs to determine which publishers in the network are promising candidates to satisfy the continuous query with appropriate documents published in the future. This publisher ranking can be decided once appropriate statistics about data sources are collected from the directory, and a ranking of the publishers is calculated based on the agent selection strategy described in Section 4.

To collect statistics about the data publishers, S needs to contact all directory agents responsible for the query terms. Thus, for each query term t_i, S computes $H(t_i)$, which is the identifier of the agent responsible for storing statistics about other publishers that publish documents containing the term t_i. Subsequently, S creates message COLLECTSTATS$(id(S), ip(S), t_i)$, and uses the function SEND() to forward the message in $O(logN)$ hops to the agent responsible for identifier $H(t_i)$. Notice that the message contains $ip(S)$, so its recipient can directly contact S.

When a agent D receives a COLLECTSTATS message asking for the statistics of term t_i, it searches its local post store to retrieve the agent list L_i of all posts of the term. Subsequently, a message RETSTATS(L_i, t_i) is created by D and sent to S using its IP found in the COLLECTSTATS message. Once S has collected all the agent lists L_i for the terms contained in q, it utilises an appropriate scoring function $score(n, q)$ to compute a agent score with respect to q, for each one of the agents n contained in L_i. Based on the score calculated for each publisher, a ranking of publishers is determined and the highest ranked agents are candidates for storing q.

Subsequently, S selects the highest ranked publishers that will index q. Thus, only publications occurring *at those publishers* will be matched against q and create appropriate notifications. Agents publishing documents relevant to q, but not indexing q, will not produce any notification for it, since they are not aware of q. Since only selected agents are monitored for publications, the publisher ranking function becomes a critical component, which will determine the final recall achieved. This ranking function is discussed in detail in the next section.

Once the agents that will store q have been determined, S constructs message INDEXQ$(id(S), ip(S), q)$ and uses the IP addresses associated with the agent to forward the message to the agents that will store q. When a publisher P receives a message INDEXQ containing q, it stores q using a local query indexing mechanism such as [35].

Filtering and agent selection are dynamic processes, therefore a periodic query repositioning, based on user-set preferences, is necessary to adapt to changes in publisher's behaviour. To reposition an already indexed query q, a subscriber would re-execute the subscription protocol, to acquire new publisher statistics, compute a new ranking, and appropriately modify the set of agents indexing q.

Publication and Notification Protocol. The publication service is employed by users that want to expose their content to the network. A publisher P utilises the directory to update statistics about the terms contained in the documents it publishes. All queries that match a published document produce appropriate notifications to interested subscribers.

According to the above, the procedure followed by P at publication time is as follows. When a document d is published by P, it is matched against P's local query database to determine which subscribers should be notified. Then, for each subscriber S, P constructs a notification message NOTIFY$(id(P), ip(P), d)$ and sends it to S using the IP address associated with the stored query. If S is not on-line at notification arrival, then P utilises function SEND() to send the message through the DHT, by using the $id(S)$ also associated with q. In this way, S will receive the message from its successor upon reconnection. Notice that agents publishing documents relevant to a query q, but not storing it, will produce no notification.

4 Publisher Ranking Strategy

To select which publishers will be monitored, the subscription protocol of Section 3.2 uses a scoring function to rank publisher agents according to quality and price. In this section we quantify these concepts, and give the rationale between our choices.

4.1 Quality vs Price

The publisher ranking strategy is a critical component, since it decides which publishers will store a continuous query. Contrary to exact information filtering, where the system would deliver all events matching a subscription, in approximate information filtering a subscriber registering with a continuous query q has to decide which publishers are the most promising candidates for satisfying q, as he will receive events that match q only from those publishers.

To make an informed selection on the publishers, a subscriber agent ranks them based on a combination of publisher quality and price quoted for the specific type of information. This combination describes the benefit/cost ratio and allows the subscriber to assign a score to every publisher. Empirical studies have shown that price and quality are the two key determinants of the consumer's choice to buy or not a product [9]. The score for each publisher is computed as follows:

$$score(P,q) = (1-\alpha) \cdot quality(P,q) - \alpha \cdot price(P,q) \quad (1)$$

In Equation 1, $quality(P,q)$ denotes how relevant P is to the continuous query q, while α is a tunable parameter that affects the balance between the importance of price over quality. The $price(P,q)$ component in Equation 1 refers to the price a publisher is quoting for published documents matching a continuous query q. The publishers are computing the price on demand according to their popularity and the popularity of their documents. The price has the same domain as quality for allowing their use within the same formula, and is recomputed whenever the popularity of the publisher changes (i.e., a new continuous query is stored at the publisher). In the experimental section we study the price in different scenarios, and show the effect on recall when the price choice is (i) random, (ii) strongly correlated with quality, and (iii) partially correlated with quality.

4.2 Calculating Publisher Quality

To assess the quality of the information producer, as required in Equation 1, the subscriber uses a combination of *resource selection* and *behaviour prediction* as shown below:

$$quality(P,q) = \gamma \cdot sel(P,q) + (1-\gamma) \cdot pred(P,q) \quad (2)$$

The functions $sel(P,q)$ and $pred(P,q)$ are scoring functions based on resource selection and publication prediction methods respectively that assign a score to a publisher P with respect to a query q. The tunable parameter γ affects the balance between authorities (high $sel(P,q)$ scores) and agents with potential to publish matching documents in the future (high $pred(P,q)$ scores). Based on these scores, a score representing the quality of a publisher is determined.

To show why an approach that scores publishers based only on resource selection is not sufficient, and to give the intuition behind publisher behaviour prediction, consider the following example. Assume an agent A_1 that is specialised and has become an authority in sports, but publishes no relevant documents any more. Another agent A_2 is not specialised in sports, but is currently crawling a sports portal, and publishing documents from it. Imagine a user who wants to stay informed about the 2011 Special Olympics, and subscribes with the continuous query *2011 Special Olympics*. If the ranking function solely relies on resource selection, agent A_1 would always be chosen to index the user's query (since it was a sports authority in the past), despite the fact that it no longer publishes sports-related documents. On the other hand, to be assigned a high score by the ranking function, agent A_2 would have to specialise in sports – a long procedure that is inapplicable in a filtering setting which is by definition dynamic. The fact that resource selection alone is not sufficient is even more evident in the case of news items. News items have a short shelf-life, making them the worst candidate for slow-paced resource selection algorithms.

Behaviour Prediction. To predict the publishing behaviour of an agent, we model IR statistics maintained in the distributed directory as time-series data and use statistical analysis tools [5] to model publisher behaviour. Time-series techniques predict future values based on past observations and differ in (i) their assumptions about the internal structure of the time series (e.g., whether trends and seasonality are observed) and (ii) their flexibility to put more emphasis on recent observations. Since the IR statistics we utilise exhibit trends, for instance, when agents successively crawl sites that belong to the same/different topics, or, gradually change their thematic focus, the employed time series prediction technique must be able to deal with trends. Furthermore, in our scenario we would like to put more emphasis on an agent's recent behaviour and thus assign higher weight to recent observations when making predictions about future behaviour. For the above reasons we chose *double exponential smoothing* (DES) as our prediction technique, since it supports decreasing weights on observed values and allows for trend in the series of data.

The function $pred(P,q)$ returns a score for a publisher P that represents the likelihood of publishing documents relevant to query q in the future. Using the DES technique described above, two values are predicted. Firstly, for all terms t in query q, we predict the value for $df_{P,t}$ (denoted as $df^*_{P,t}$), and use the difference (denoted as $\delta(df^*_{P,t})$) between the predicted and the last value obtained from the directory to calculate the score for P (function $\delta()$ stands for difference). Value $\delta(df^*_{P,t})$ reflects the number of relevant documents that P will publish in the next period. Secondly, we predict $\delta(cs^*)$ as the difference in the collection size of agent P reflecting the agent's overall expected future publishing activity. We thus model two aspects of the publisher's behaviour: (i) its potential to publish relevant documents in the future (reflected by $\delta(df^*_{P,t})$), and (ii) its overall expected future publishing activity (reflected by $\delta(cs^*)$). The time series of IR statistics that are needed as an input to our prediction mechanism are obtained using the distributed directory. The predicted behaviour for agent P is quantified as follows:

$$pred(P,q) = \sum_{t \in q} \log\left(\delta(df^*_{P,t}) + \log\left(\delta(cs^*_P) + 1\right) + 1\right) \tag{3}$$

In the above formula, the publishing of relevant documents is more accented than the dampened publishing rate. If an agent publishes no documents at all, or, to be exact, the prediction of $\delta(df^*_{P,t})$ or $\delta(cs^*_P)$ is 0 then the $pred(P,q)$ value is also 0. The addition of 1 in the log formulas yields positive predictions and avoids $\log(0)$.

Resource Selection. The function $sel(P,q)$ returns a score for a publisher P and a query q, and is calculated using standard resource selection algorithms from the IR literature, such as tf-idf based methods, CORI or language models (see [26] for an overview). Using $sel(P,q)$ we identify authorities specialised in a topic, which, as argued above, is not sufficient for our IF setting. In our implementation we use an approach based on document frequency (df), and maximum term frequency (tf^{max}). The values of $sel(P,q)$ for all query terms t are aggregated as follows:

$$sel(P,q) = \sum_{t \in q} \beta \cdot \log\left(df_{P,t}\right) + (1-\beta) \cdot \log\left(tf^{max}_{P,t}\right) \tag{4}$$

The value of the parameter β can be chosen between 0 and 1 and is used to emphasise the importance of df versus tf^{max}. Experiments with resource selection have shown that β should be set around 0.5.

4.3 Economic Modelling of ABIS

In this section we analyse the economic modeling of ABIS and review the basic assumptions and expectations from such a modelling.

Usefulness of the information goods received by a subscriber, is a qualitative criterion, that is difficult to model. In ABIS we model usefulness by matching interest, i.e., by assuming that all received documents relevant to the requested topic are useful to the subscriber, and do not discuss issues such as novelty and coverage of information, or user effort. In our modelling, after a subscriber acquires a history of transactions with certain publishers it develops an affect for some of the publishers. Affect can be modelled in various ways, depending on the task at hand, and can be either positive or negative (as in e.g., [10] where affect causes a “preference shock" to the consumers that buy only from a certain manufacturer). In ABIS, an information consumer does not know the quality of the information goods, but he uses the affect developed from previous transactions to approximate it. Subsequently, he compares the values of information quality to the expected values and update its affect [25].

The costs in ABIS are results of agent actions [18], such as transactions (e.g., unsubscribing from a publisher and subscribing to another, changing a submitted query), network communication, and use of common infrastructure (e.g., the directory service). Since each agent may play a dual role both as a publisher and a subscriber, it will naturally try to maximise his revenue, and the utility of the received resources, while minimising expenses that occur due to publication or subscription actions. In general, the information market in the ABIS system is not a pure competitive market [38] since the subscribers do not know in advance the exact quality of the information they are buying. The ABIS system resembles the modelling of a team of sales people [37]. In this model agents would try to collaborate with others in order to get their expertise for a (cross/up) sale. After deciding which agents to collaborate with, it will be possible to model the gap between the initial expectations and the actual actions of the agent. In [12] it is shown that this gap is smaller in a competitive relationship compared to that of a cooperative relationship. As in many cooperative environments each agent usually retains its connections with the other agents, while also being free to explore new mutually beneficial connections.

The main goal of this agent-based modelling is to study the influence of the cost component on the quality of received resources, study the interactions between agents that are trying to maximise the benefits of information flow, and gain insights about the activity and the behaviour of the publishers and subscribers.

5 Experimental Evaluation

In this section we present our findings regarding the introduction of cost in information goods, and how it affects the effectiveness of an information filtering system. We study

the behaviour of the ABIS system using different publishing scenarios, while varying the correlation between price, quality and customer demand.

5.1 Experimental Setup

To conduct each experiment described in the next sections the following steps are executed. Initially the network is set up and the underlying Distributed Hash Table (DHT) is created. Then, subscribers use the ranking function based on resource selection, predicted publishing behaviour and cost of information to decide which are the best publishers subscribe to. Subsequently, they utilise the subscription protocol described in detail in Section 3.2 to subscribe to the selected publishers. Once the queries are stored, the documents are published to the network and at certain intervals (called *rounds*) queries are repositioned, and new documents are published.

Evaluation Metrics. To measure the effect of cost in information filtering, and compare between cases of IF with and without monetary flow in ABIS, we utilise the following metrics:

- **Messages.** We measure the number of directory, subscription and notification messages in the system to perform the filtering task at hand.
- **Recall.** We measure recall by computing the *ratio* of the total number of notifications received by subscribers to the total number of published documents matching subscriptions. In experiments we consider the *average recall* computed over *all* rounds (i.e., for the complete experiment).
- **Ranking.** We use an extension of Spearman's footrule distance to compare rankings of publishers calculated by subscribers. This metric allows us to compare two different publisher rankings by calculating the distance between the elements in two ranking lists. In our extension of Spearman's metric, if an element from list A is not present in list B, it is considered as being in the last available position in B.

System Parameters. There is a number of system parameters that regulate agent behavior and had to be determined and set. Due to space considerations the procedure and experimentation of finding the optimal values for these parameters is omitted and the interested reader is referred to [44]. One such key parameter is the percentage of the available publishers that a subscriber can follow. When all publishers are monitored, then recall is 100%, and our approach degenerates to exact filtering. Exact information filtering will always give the best result with regard to recall, but also incur high message traffic to the system and cost to the subscriber. On the other hand, when using a random selection of publishers, monitoring $k\%$ of publishers will typically result in recall around $k\%$. To achieve higher than $k\%$, the publisher ranking strategy presented in Section 4 is employed. Additionally, parameter γ in Equation 2 controls the balance between resource selection and behavior prediction; a value of γ close to 0 emphasises behavior prediction, while values close to 1 emphasise resource selection. In previous work [44], we determined that monitoring around 10% of publishers, and setting the value of γ to 0.5 represents a good trade-off between recall and message overhead. Additionally, both coefficients used in the double exponential smoothing were set to 0.5, as in [45].

Finally, for deciding the budget per agent, we relied to studies on budget distribution and spending for a variety of cases, ranging from family budgets to consumer budgets [20,6]. The main conclusions drawn from these studies are that (i) budget distribution follows a power law, with a small percentage of families/consumers having a high yearly budget, and a large percentage of the families being in the (long) tail of the distribution, with a low budget, and (ii) the percentage of the income spend on (information) goods does not vary with the budget. According to [3] and the above remarks, we divided the agents into three classes: low budget agents, average budget agents, and high budget agents. 60% of the agents are part of the low budget class, 30% of the agents have an average budget, and 10% belong to the high budget agent class. Subsequently, we experimentally computed a budget that would allow the information consumers to subscribe to all top-k publishers, and allowed the low budget agents to have 60%, the medium budget agents to have 80% and the high budget agents to have 120% of this ideal budget.

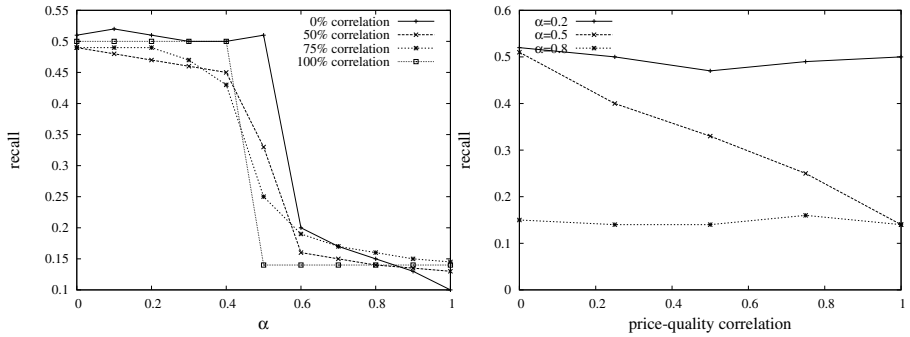
Documents and Queries. The document collection contains over 2 million documents from a focused Web crawl categorised in one of ten categories: *Music*, *Finance*, *Arts*, *Sports*, *Natural Science*, *Health*, *Movies*, *Travel*, *Politics*, and *Nature*. The overall number of corpus documents is $2,052,712$. The smallest category consists of $67,374$ documents, the largest category of $325,377$ documents. The number of distinct terms after stemming adds up to $593,876$.

In all experiments, the network consists of $1,000$ agents containing 300 documents each in their initial local collection. Each agent is initialised with 15% random, 10% non-categorised, and 75% single category documents, resulting in 100 specialised agents for each category. Using the document collection, we construct continuous queries containing two, three or four query terms. Each of the query terms selected is a strong representative of a document category (i.e., a frequent term in documents of one category and infrequent in documents of the other categories). Example queries are *music instrument*, *museum modern art*, or *space model research study*.

5.2 Varying the Price-Quality Correlation

In this experiment we aimed at observing the behaviour of the system when we varied the correlation between the price and quality of publisher. In this experiment, 100% correlation between price and publisher quality, means that the better the quality of the publisher, the higher the price it charges for publications. In this case quality can be easily forecasted and consumers will know that the information goods will be expensive but useful to them. The other extreme in this correlation is when prices have no (0%) correlation with quality, and are chosen randomly. In the case of 75% (respectively 50% and 25%) correlation between price and quality, we modelled the correlation, as the likelihood that a publisher sells 25% (respectively 50% and 75%) of the information goods underpriced up to 20% of the initial value, and 75% (respectively 50% and 25%) overpriced up to 10% of the initial value.

In Figure 1(a) the achieved recall of the system for different values of α and different price-quality correlations is shown. The first observation emanating from this graph is that the introduction of a price component reduces the observed recall of the system (notice that recall has the highest values for $a = 0$, i.e., not pricing involved in the



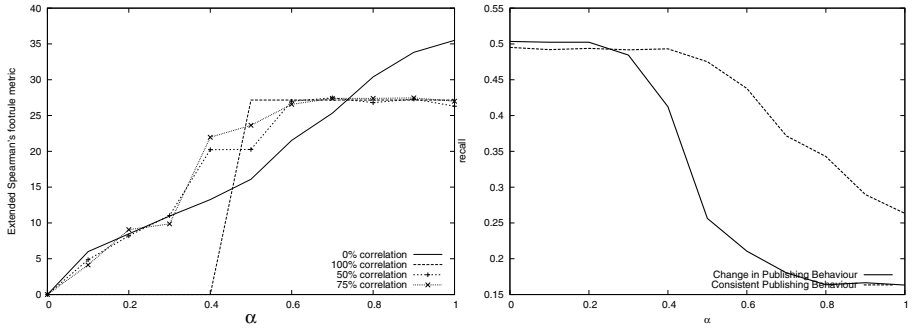
(a) (b)

Fig. 1. Recall against α and price-quality correlations

ranking of publishers). This is an important result, showing that when information publishers charge for information, consumers trade quality subscriptions for cheaper ones. Notice also that in all cases of price-quality correlation, recall is retained high, as long as it plays the most important role in the ranking ($\alpha < 50\%$). This was also expected, since when price is of importance, consumers will choose cheaper publishers, leading to a reduction in the observed recall. Additionally, when the price is the only ranking criterion for information consumers ($a = 1$), recall is close to that of a random choice of publishers (remember that agents monitor only 10% of publishers in the system).

Another observation is that the correlation between the price set by the publisher and its actual quality, plays (as expected) an important role only when price and quality are equally important. When one of the two components becomes dominant in the ranking function, it outweighs the effect of the other. This is also in accordance with our expectations, since when price comes into the picture, quality is sacrificed to reduce costs, or increase the received publications. These observations are best shown in Figure 1(b) where recall for varying the value of price-quality correlation and three different values of α is presented. Finally, notice that the small variation in the observed recall and agent behavior between different price-quality correlations, is also partly due to the modelling of ABIS as a closed system, where monetary flow is limited through the budgets of the agents, since no new wealth is produced.

Figure 2(a) shows the difference in publisher rankings when varying α and for different price-quality correlations. The difference in the ranking of publishers is measured using an extension of Spearman's footrule metric. To produce a point in the graph we compare the list of publishers ranked by a subscriber when no cost is introduced, and the same list when we introduce cost with the given value for α. This is performed for all the subscribers in the system, and the average metric is calculated. The first observation emanating from the graph is that for $\alpha = 0$, all price-quality correlations are naturally zero, since no cost is associated with the information goods, and thus the lists compared are identical. Another observation, is that when α is increasing, i.e., price becomes more important in the ranking process, Spearman's metric increases too, as publishers with high quality get lower positions in the ranking, while publishers with lower quality (but cheaper) are ranked high. Additionally, notice that the difference in



(a) Difference in publisher ranking (b) Recall for different publishing scenarios

Fig. 2. Publisher rankings and behaviors against α

recall observed in Figure 1, is also partially depicted here as difference in the ranking of the information producers. Finally, when the price of a publisher has no association with the quality of its documents (random price setting), the difference in the ranking of publishers is about 40% higher, than the case of price and quality being correlated (leftmost points in the graph).

5.3 Varying the Publishing Behaviour

In this section we look into recall and how this is affected by different publishing behaviours, when varying the importance of information cost in the system. The publishing behavior of agents is modelled using two different scenarios: *consistent publishing* and *category change*, that represent the two extremes of publishing behaviours.

Consistent publishing. In the consistent publishing scenario, the publishers maintain their specialisation, by disregarding market conditions, even if this results in very low revenue.

Category change. In the category change scenario, publishers change their topic of specialisation over time based on changes in consumer behaviour, revenue and market conditions. In this scenario, a publishing agent initially publishes documents from one category, and switches to a different category after a number of rounds, to simulate changes in portfolio contents or business strategies.

As we can observe in Figure 2(b), in both scenarios, the system reaches the highest recall when no price component is added (leftmost point in the graph), while as cost of information gains importance, the observed recall drops, since agents seek for cheaper publishers. A second observation, is that the consistent publishing scenario is less affected by the introduction of the price component, and achieves significantly higher recall as α increases. This happens because in this scenario, publishers have build up an expertise, and since this expertise is not changed, the quality component increases, leading to the ranking of these publishers high in the list. Contrary, when publishers change their area of focus, the observed recall of the system falls, since subscribers are

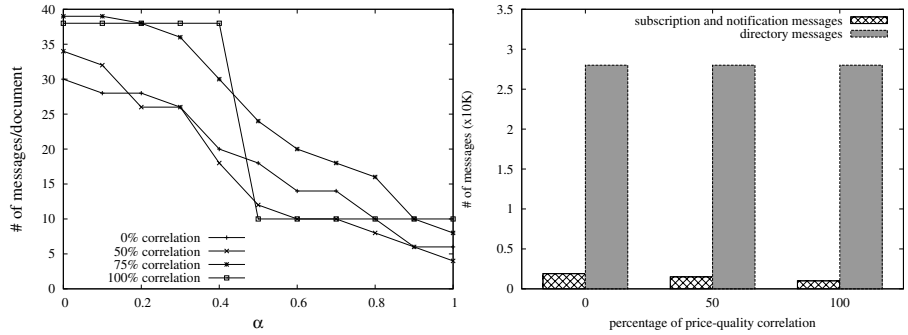


Fig. 3. Message traffic against α and price-quality correlations

not able to correlate price and quality for the publishers (see also Section 5.2). This observation resembles the case of companies that have to allocate a marketing budget to convince consumers about a new product. Here, the publishers change their publishing behaviour to sell a new product (i.e., a new topic) and the old customers walk away, resulting in recall reduction.

5.4 System Performance

In this series of experiments we targeted the system performance in terms of message traffic. In Figure 3(a) we present the message traffic per agent (subscription and notification messages) incurred in the system when varying α. In this graph we see that the number of messages per agent is reduced, as the price component is emphasised. This can be explained as follows. As subscribers utilise the price component to rank publishers, they choose publishers of lower quality and price. This, as we also observed in the previous sections, results in a reduction in the observed recall, since subscribers receive less notifications, as they subscribe to non-expert publishers. On the other hand expert publishers have a smaller customer base, and are thus forced to notify fewer subscribers. The interested reader is also referred to [44], where we demonstrated that recall and message traffic are interconnected.

Figure 3(b) demonstrates the total amount traffic observed in the system, and how this traffic is split in the various message categories, as the price-quality correlation is varied. As expected the directory traffic dominates the messaging load of the system, as necessary messages with agent statistics and prices are disseminated. Notice that directory traffic is not affected by the correlation between price and quality, since the publishers are responsible for updating their publication statistics and prices, regardless of the size of their customer base. Finally, notice that the number of subscription and notification messages is slightly affected from the price-quality correlation, as quality publishers widen their customer base with more subscribers.

5.5 Summary of Results

The experiments presented in this section, show the behaviour of a IF system when a price component is introduced in the selection process of publishers. To the best of

our knowledge these are the first results that connect recall and message traffic with the cost component, and put economic modelling in the picture of distributed IF. Our findings show that when introduced, the price component affects the average recall of the system, since it outweighs quality in the ranking of publishers. Our experiments showed that the price component should participate in publisher ranking with no more than 10-20% of the total score, to avoid loss in observed recall. Additionally, we showed that adding a price component in such a system, reduces message traffic, as (i) this is directly connected to recall, and (ii) agents avoid costly actions like frequent document publications, and query repositioning. Thus, pricing information goods in distributed settings should be carried out carefully, to avoid user dissatisfaction due to reduced flow of relevant documents.

6 Conclusions and Outlook

In this work we have defined an architecture and the associated protocols to achieve distributed agent-based approximate IF, and introduced a novel publisher selection mechanism that ranks monitored information producers according to their expertise, their predicted publishing behavior (based on time-series analysis of IR metrics) and the price of the information goods they publish. We have showed that approximate IF is an efficient and effective alternative to the exact IF paradigm, as it manages to trade recall for low message overhead, while providing an interesting business model. We are currently porting our implementation to PlanetLab to conduct more extensive experimentation, and adding new features such as monitoring of monetary flow.

References

1. Aekaterinidis, I., Triantafillou, P.: PastryStrings: A Comprehensive Content-Based Publish/Subscribe DHT Network. In: ICDCS (2006)
2. Bender, M., Michel, S., Parkitny, S., Weikum, G.: A Comparative Study of Pub/Sub Methods in Structured P2P Networks. In: Moro, G., Bergamaschi, S., Joseph, S., Morin, J.-H., Ouksel, A.M. (eds.) DBISP2P 2005 and DBISP2P 2006. LNCS, vol. 4125, pp. 385–396. Springer, Heidelberg (2007)
3. Breker, L.P.: A survey of network pricing schemes. In: Theoretical Computer Science (1996)
4. Carzaniga, A., Rosenblum, D.-S., Wolf, A.: Design and Evaluation of a Wide-Area Event Notification Service. In: ACM TOCS (2001)
5. Chatfield, C.: The Analysis of Time Series - An Introduction. CRC Press (2004)
6. DeLong, J.B.: Six Families Budget Their Money. In: Lecture notes for American Economic History, University of California at Berkeley (2008)
7. Demetriades, I., Lee, T.Y., Moukas, A., Zacharia, G.: Models for pricing the distribution of information and the use of services over the Internet: A focus on the capital data market industry (1998), http://web.mit.edu/ecom/www/Project98/G12/
8. Drosou, M., Stefanidis, K., Pitoura, E.: Preference-aware publish/subscribe delivery with diversity. In: DEBS (2009)
9. Dumrogsiri, A., Fan, M., Jain, A., Moinzadeh, K.: A supply chain model with direct and retail channels. European Joumal of Operational Research (2008)
10. Faig, M., Jerez, B.: Inflation, Prices, and Information and Competitive Search. Journal of Macroeconomics (2006)

11. Feldman, M., Lai, K., Chuang, J., Stoica, I.: Quantifying Disincentives in Peer-to-Peer Networks (2003), http://www.cs.berkeley.edu/~istoica/papers/2003/discincentives-wepps.pdf
12. Forker, L., Stannack, P.: Cooperation versus Competition: do buyers and suppliers really see eye-to-eye? European Journal of Purchasing and Supply Management (2000)
13. Fuqua, A., Ngan, T.-W.J., Wallach, D.: Economic Behavior of Peer-to-Peer Storage Networks. In: Economics of Peer-to-Peer Systems (2003)
14. Gedik, B., Liu, L.: PeerCQ: A Decentralized and Self-Configuring Peer-to-Peer Information Monitoring System. In: ICDCS (2003)
15. Gupta, A., Sahin, O.D., Agrawal, D., El Abbadi, A.: Meghdoot: Content-Based Publish/Subscribe Over P2P Networks. In: Jacobsen, H.-A. (ed.) Middleware 2004. LNCS, vol. 3231, pp. 254–273. Springer, Heidelberg (2004)
16. Hassan, A., Elie, K.: MAWS: A platform-independent framework for Mobile Agents using Web Services. In: JPDC (2006)
17. Idreos, S., Koubarakis, M., Tryfonopoulos, C.: P2P-DIET: One-Time and Continuous Queries in Super-Peer Networks. In: Hwang, J., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K. (eds.) EDBT 2004. LNCS, vol. 2992, pp. 851–853. Springer, Heidelberg (2004)
18. Johansson, B., Persson, H.: Self-organised adjustments in a market with price-setting firms. In: Chaos, Solitons and Fractals (2003)
19. West, Jr., L.A.: Private Markets for Public Goods: Pricing Strategies of Online Database Vendors. In: Journal of Management of Information Systems (2000)
20. Kennickell, B.B.A., Moore, K.: Recent Changes in U.S. Family Finances: Evidence from the 2001 and 2004 Survey of Consumer Finances (2006)
21. Khouja, M., Hadzikadic, M., Rajagopalan, H., Tsay, L.: Application of complex adaptive systems to pricing reproductible information goods. Decision Support Systems (2007)
22. Koubarakis, M., Koutris, T., Tryfonopoulos, C., Raftopoulou, P.: Information Alert in Distributed Digital Libraries: The Models, Languages, and Architecture of DIAS. In: Agosti, M., Thanos, C. (eds.) ECDL 2002. LNCS, vol. 2458, p. 527. Springer, Heidelberg (2002)
23. Koubarakis, M., Tryfonopoulos, C., Idreos, S., Drougas, Y.: Selective Information Dissemination in P2P Networks: Problems and Solutions. In: SIGMOD Record (2003)
24. Koubarakis, M., Tryfonopoulos, C., Raftopoulou, P., Koutris, T.: Data Models and Languages for Agent-Based Textual Information Dissemination. In: Klusch, M., Ossowski, S., Shehory, O. (eds.) CIA 2002. LNCS (LNAI), vol. 2446, p. 179. Springer, Heidelberg (2002)
25. Li, C., Singh, M., Sycara, K.: A Dynamic Pricing Mechanism for P2P Referral Systems. In: AAMAS (2004)
26. Nottelmann, H., Fuhr, N.: Evaluating Different Methods of Estimating Retrieval Quality for Resource Selection. In: SIGIR (2003)
27. Pietzuch, P., Bacon, J.: Hermes: A Distributed Event-Based Middleware Architecture. In: Proceedings of the International Workshop on Distributed Event-Based Systems, DEBS (July 2002)
28. Raje, R., Qiao, M., Mukhopadhyay, S., Palakal, M., Peng, S., Mostafa, J.: Homogeneous Agent-Based Distributed Information Filtering. In: Cluster Computing (2002)
29. Ratnasamy, S., Francis, P., Handley, M., Karp, R.M., Shenker, S.: A Scalable Content-Addressable Network. In: SIGCOMM (2001)
30. Rowstron, A., Kermarrec, A.-M., Castro, M., Druschel, P.: Scribe: The Design of a Large-scale Event Notification Infrastructure. In: Crowcroft, J., Hofmann, M. (eds.) COST264 Workshop (2001)
31. Stoica, I., Morris, R., Karger, D.R., Kaashoek, M.F., Balakrishnan, H.: Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In: SIGCOMM (2001)

32. Tang, C., Xu, Z.: pFilter: Global Information Filtering and Dissemination Using Structured Overlay Networks. In: FTDCS (2003)
33. Tryfonopoulos, C., Idreos, S., Koubarakis, M.: LibraRing: An Architecture for Distributed Digital Libraries Based on DHTs. In: Rauber, A., Christodoulakis, S., Tjoa, A.M. (eds.) ECDL 2005. LNCS, vol. 3652, pp. 25–36. Springer, Heidelberg (2005)
34. Tryfonopoulos, C., Idreos, S., Koubarakis, M.: Publish/Subscribe Functionality in IR Environments using Structured Overlay Networks. In: SIGIR (2005)
35. Tryfonopoulos, C., Koubarakis, M., Drougas, Y.: Filtering Algorithms for Information Retrieval Models with Named Attributes and Proximity Operators. In: SIGIR (2004)
36. Tryfonopoulos, C., Zimmer, C., Weikum, G., Koubarakis, M.: Architectural Alternatives for Information Filtering in Structured Overlays. In: Internet Computing (2007)
37. Üstüner, T., Godes, D.: Better Sales Networks. In: Harvard Business Review (2006)
38. Varian, H.R.: Pricing Information Goods. In: Research Libraries Group Symposium, Harvard Law School (1995)
39. Varian, H.R.: Pricing Electronic Journals. D-Lib Magazine (1996)
40. Varian, H.R.: Versioning Information Goods (1997), `http://people.ischool.berkeley.edu/~hal/Papers/version.pdf`
41. Varian, H.R.: Buying, Sharing and Renting Information Goods. Journal of Industrial Economics (2000)
42. Zhang, R., Hu, Y.C.: HYPER: A Hybrid Approach to Efficient Content-Based Publish/Subscribe. In: ICDCS (2005)
43. Zhu, Y., Hu, Y.: Ferry: A P2P-Based Architecture for Content-Based Publish/Subscribe Services. In: IEEE TPDS (2007)
44. Zimmer, C., Tryfonopoulos, C., Berberich, K., Koubarakis, M., Weikum, G.: Approximate Information Filtering in Peer-to-Peer Networks. In: Bailey, J., Maier, D., Schewe, K.-D., Thalheim, B., Wang, X.S. (eds.) WISE 2008. LNCS, vol. 5175, pp. 6–19. Springer, Heidelberg (2008)
45. Zimmer, C., Tryfonopoulos, C., Berberich, K., Weikum, G., Koubarakis, M.: Node Behavior Prediction for LargeScale Approximate Information Filtering.In: LSDS-IR (2007)
46. Zimmer, C., Tryfonopoulos, C., Weikum, G.: MinervaDL: An Architecture for Information Retrieval and Filtering in Distributed Digital Libraries. In: Kovács, L., Fuhr, N., Meghini, C. (eds.) ECDL 2007. LNCS, vol. 4675, pp. 148–160. Springer, Heidelberg (2007)