X

Service Ratio-Optimal, Content Coherence-Aware Data Push Systems

Advertising new information to users via push is the trigger of operation for many contemporary information systems. Furthermore, passive optical networks are expected to extend the reachability of high quality push services to thousands of clients. The efficiency of a push service is the ratio of successfully informed users. However, pushing only data of high popularity can degrade the thematic coherency of the content. The present work offers a novel, analysis-derived, tunable way for selecting data for push services. The proposed scheme can maximize the service ratio of a push system with regard to data coherence constraints. Extensive simulations demonstrate the efficiency of the scheme compared to alternative solutions. The proposed scheme is the first to tackle the problem of data coherence-aware, service ratio optimization of push services.

Categories and Subject Descriptors: H.3.3 [**Information Systems**]: Information Filtering; C.2.0 [**Computer-Communication Networks**]: Data Communications

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Data push, data selection, service ratio, coherence

ACM Reference Format:
2014. Service Ratio-Optimal, Content Coherence-Aware Data Push Systems. *ACM Trans. Manag. Inform. Syst.* X, X, Article X (March 2014), 23 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Data push services are integral parts of many modern information services [Peng et al. 2008]. The abundant online data and services would remain unknown and unused without proper promotion. A crucial aspect of push is the selection of data for forwarding to the clients out of a given pool. A successful push service must primarily select and forward data that maximize its service ratio, since it translates to more subscriptions and economic viability. Nonetheless, blind service ratio optimization without regard to the thematic coherence of the pushed data is rarely an acceptable solution [Adomavicius and Zhang 2012]. The present work proposes a novel data selection algorithm for push systems which offers maximum service ratio for a freely predefined level of data coherency. To best of the authors' knowledge, the present work is the first to address this issue.

Prominent application examples of data push systems are the IPTV services. Such a service has wide population coverage, being typically implemented over a cost-efficient passive optical network. Much like classic TV, an IPTV channel offers several types of content, including news briefings, pastime, education, athletics, financing, etc. As any push service, IPTV channels do not rely on client queries, but rather on client preference statistics. The statistics refer to the popularity distribution of the available shows (data items) or their criticality, expressing the impatience of the viewers for a given show. Exact ways of deriving these statistics are studied as a separate field of study and may include selective demographics (polling of volunteers), monitoring of

social media or incurred network traffic [Liaskos and Papadimitriou 2012; Nicopolitidis et al. 2002]. The IPTV service then selects a subset of the available shows and creates a push time-schedule. Such a schedule is primarily designed to maximize the rate of shows successfully pushed within their statistical deadlines (service ratio optimization). This approach maximizes the number of subscribed users and ensures the long-term economic viability of the IPTV service. Past and recent studies have addressed the issue of service ratio optimization [Balli et al. 2007; Jiang and Vaidya 1999; Raissi-Dehkordi and Baras 2007; Liaskos et al. 2013]. However, no study has taken into account the thematic similarity of data selected for push (coherence). Practically, it is doubtful that any viewer would be interested in a channel that broadcasts financing news and animated shows, despite what their popularity statistics may imply. A certain degree of coherence must be retained inside the schedule.

The present work offers a data selection scheme that maximizes the service ratio while keeping the coherence of the data bounded. The methodology of the study acknowledges that the two goals are statistically uncorrelated. In other words, a service ratio-optimal subset of IPTV shows may or may not be coherence-optimal, equiprobably. The ideal solution would be to produce all item subsets that yield a pre-defined level of coherence, and then simply select the one promising the maximum service ratio. However, this is a known NP-Hard problem, even at its first stage [Schoning 1999]. Another approach would be to apply the goals sequentially, deriving the most coherent subsets first and then re-selecting for high service ratio (or vice-versa). Nonetheless, this approach too has been shown to produce unsatisfactory results, at least in terms of service ratio [Liaskos et al. 2013; 2012]. In light of these facts, the present study proposes a clustering-based selection process with parametric bias. Setting this bias to a value between 0 and 100% gradually steers the selection process from coherence-oriented to service-ratio oriented. The IPTV service can then select the intermediate point of operation fit for its case. The tunable clustering process is enabled by the core-contribution of the present study, which represents the service ratio optimality goal as a virtual "distance" between any two data items. This approach also extends the applicability of existing clustering algorithms to the problem of data selection for push systems. Simulations show that the proposed scheme: i) is highly efficient with regard to existing solutions, ii) is able to seamlessly combine different selection processes (e.g. linear and non-linear ones), iii) can be incorporated to a wide variety of existing, generic clustering algorithms. Simulations employ real and synthetic data.

The remainder of this paper is organized as follows. Related studies are presented in Section 2. The system model and prerequisites are given in Section 3. The proposed clustering scheme is detailed in Section 4. Incorporation to well-known clustering algorithms is studied in Section 5. Section 6 evaluates the analytical results through simulation and comparison to alternative approaches. The conclusion follows in Section 7.

2. RELATED WORK

Advertising/pushing data and services to users is crucial to the operation of many modern information systems. Emergency response systems operate by pushing messages to responders [Valecha et al. 2013]. Interactive TV broadcasting is another, multimedia-push example, either in its wireless form or the more recent IPTV over passive optical networks [Ikeda et al. 2007]. The latter is adopted as a persistent example throughout the study. Data push in social networks is also an imporant topic of research [Rui and Whinston 2012].

Two types of advertising are defined based on the data dissemination scheme. Pull-based schemes assume duplex communication between the client and the server. Each client posts explicit queries and the server serializes the answers in an optimal way

[Jianliang Xu et al. 2006]. While personalized data advertising is at first ideal, it raises issues of privacy, user consent, security and data management [Unni and Harmon 2007].

Data push, which is studied in this paper, is a risk-free, cost-effective alternative. It enforces one-way communication: a client does not post queries to the server, but rather waits until an item of interest appears in the pushed data stream (hidden query). In this case, the server collects statistics only per data item, which suffice for optimal operation. A typical statistic is the data popularity, i.e. the percentage of hidden queries that refer to a given data item.

Early studies on data push attempted to minimize the mean service time of the hidden queries. Authors in [J. Gecsei 1983] showed that the optimal schedule in this case is periodic. The time interval between two consecutive item occurrences in the schedule must be constant. These intervals are typically expressed in ratio form and their values depend on the collected statistics. While the definition of the push intervals is generally tractable, periodic serialization is NP-Hard. Service preemption has been shown to be beneficial under circumstances [Serpanos 2004] and [Liaskos et al. 2011] presented a low-complexity serializer that achieved nearly-optimal results. The NP-Hardness of periodic serialization introduced several heuristic alternatives. The Broadcast Disks model assumes a virtual system of disks rotating around a common axis [Acharya et al. 1995]. The disks represent items with similar popularity. A set of imaginary heads reads and serializes data from the disks, thus forming the schedule. Studies have addressed the issues of multichannel data push [Zheng et al. 2005], pull-push hybrid data dissemination [Kim and Kang 2010; Kang et al. 2007] and data indexing for saving power at the receiving devices by shutting down the network interface for prolonged intervals [Chih-Lin Hu and Ming-Syan Chen 2009].

More recent studies can be categorized by their functionality layer orientation. At physical layer, push was combined with network coding techniques in passive optical infrastructures in [Fouli et al. 2011], essentially doubling the throughput of the scheme. Similar effects are produced in wireless infrastructures as well [Zhan et al. 2011]. At application layer, the study of [Liaskos et al. 2012] allowed for push schedule optimization under multiple criteria (multiple metrics, multiple constraints). It was shown that optimal push schedules are periodic even in the presence of complex optimization criteria. Finally, adaptive collection of data statistics by learning automata was studied by [Kakali et al. 2011].

Selecting data to push to an audience is also related to Group Recommendation Systems (GRS). By definition, Push systems serve a common stream of data to a group of clients, while GRSs aim at identifying data that are requested by a group as a whole, rather than by individuals [Jameson and Smyth 2007; Masthoff 2004]. Exemplary GRS applications include recommending news [Cleger-Tamayo et al. 2012; Das et al. 2007; Lee and Park 2007], movies [Al-Shamri and Bharadwaj 2007; Miller et al. 2003; Rattanajitbanjong and Maneeroj 2009], TV programs [O'Connor et al. 2001; Yu et al. 2006], music [Crossen et al. 2002], web pages [Pizzutilo et al. 2005], tourism [Garcia et al. 2011] and products on Amazon.com. GRSs receive the individual preferences of many users and produce a single, aggregate recommendation list [Jameson and Smyth 2007; Berkovsky and Freyne 2010]. Different aggregation functions such as average, least misery (i.e., using the minimum of individual ratings), most pleasure (i.e., using the maximum of ratings) have been proposed [Baltrunas et al. 2010; Masthoff 2004]. Aggregation can also be applied over existing recommendation lists, instead of individual preferences, e.g., by using ensemble-based systems [Plumbaum et al. 2010; Polikar 2006]. Using these approaches, GRSs are able to re-purpose per-client recommendation techniques for serving arbitrary client groups. Push systems need the output of a GRS in order to operate. In essence, the studied push systems an-

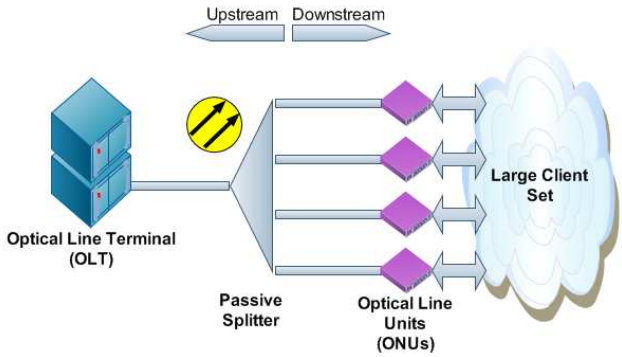


Fig. 1: Overview of a long-reach Passive Optical Network (PON), a typical example of a system operating by the broadcast and select logic. In the downstream direction, every data item pushed by the OLT (push server) reaches every line unit (ONU) and its connected clients. The upstream is used for collecting user statistics.

swer the question "when to broadcast a data item" (scheduling), while GRS supply the answer to "how popular is a data item" (statistics). The proposed method for data selection in push systems lies between the scheduling process and the GRS output. The proposed method answers the question "which of the data items should be broadcast" in order to maximize the service ratio of the push system, while keeping the coherence of the stream in check.

The present work extends the study of [Liaskos et al. 2012], which employed clustering schemes for data selection in push systems. The study showed that focusing on a very small portion of the original data pool maximizes the service ratio of the client queries. The remaining data items are not pushed at all. However, the study discarded any concern of data coherency, focusing solely on the service ratio. [Liaskos et al. 2012] and [Liaskos et al. 2013] followed an alternative, analytical approach for a single or multiple available push streams. Still, all studies could lead to the selection of data items that are promising in terms of service ratio, but are unrelated otherwise. To the best of the authors knowledge, balancing service ratio prospects and data coherency is studied for the first time in the context of push systems.

3. SYSTEM MODEL AND PREREQUISITES

Figure 1 presents an overview of the studied system. A passive optical network architecture is used for implementing a push application, such as an advertising service or a TV over fiber application. PONs are employed for providing high quality services to thousands of users in a cost effective manner. Providing 10 Gbps downstream and 2.5 Gbps upstream, next generation PONs (XGPON) can exemplary serve more than 100 HDTV channels in a multimedia push application [Ikeda et al. 2007]. The topology comprises a central push server (OLT), one ore more passive splitters (simple optical prisms) and a set of up to 1024 line units (ONUs) [ITU-T 2010]. The upstream and downstream channels occupy separate wavelengths. While the downstream is collision-free, the upstream slots of the ONUs require special handling. The OLT regulates the ONU upstream slots via a dynamic TDMA scheme [Man-Soo 2012]. The ONUs act as interfaces for any other user access technology (e.g. 3/4G, WiMAX, WiFi, Ethernet, ADSL). Thus, each PON can easily serve several thousands of users, which makes it ideal for push applications. The downstream direction is used for data push, while statistics are collected via the upstream.

The setup assumes that the push server, located at the OLT, hosts or has access to a data pool of $i = 1 \dots N$ items. The nature of the data items is irrelevant. The study exemplary assumes multimedia shows in an IPTV program. Each item has a fixed size l_i. We assume that the OLT has already collected the per item statistics using any of the existing schemes (e.g. [Papadimitriou et al. 2002]). The following, standard statistics are considered [Jiang and Vaidya 1999]:

— The popularity of an item, p_i, expressing the probability that a given hidden query is associated with item i at any time.
— The criticality, τ_i, of item i. Related studies model the clients' attention span as an exponential probability distribution [Jiang and Vaidya 1999],

$$P_a(d) = \tau_i \cdot e^{-\tau_i \cdot d} \tag{1}$$

which expresses the probability of a client waiting for time d before abandoning a hidden query. The item-specific attribute τ_i regulates the steepness of the distribution. High values correspond to more time-sensitive items.

The push server at the OLT selects a subset of the available items and begins to push them to the clients periodically for an interval of total duration L. Each selected item appears v_i times in this interval. In other words, it holds:

$$L = \sum_{i=1}^{m} v_i \cdot l_i \tag{2}$$

where m is the number of selected items, while the interval between two appearances of item i is:

$$w_i = \frac{L}{v_i} \tag{3}$$

The push server must not only select the proper data subset, but also set the v_i quantities in a way that optimizes the service ratio (percentage of non-abandoned hidden queries). The study of [Liaskos et al. 2012] showed analytically that:

— The data selection must obey to the condition:

$$\sum_{i=1}^{m} \frac{\tau_i \cdot l_i}{1 + W\left[\frac{1}{e}\left(\frac{min\left\{\frac{p_i}{\tau_i \cdot l_i}\right\}}{\frac{p_i}{\tau_i \cdot l_i}} - 1\right)\right]} > -1 \tag{4}$$

— The occurrences of the selected items in the periodic push must be set as:

$$v_i = \frac{-L \cdot \tau_i}{1 + W\left(\frac{-p_i + c \cdot \tau_i \cdot l_i}{e \cdot p_i}\right)} \tag{5}$$

where $W(x) < -1$ is the Lambert-W function[1] [Corless et al. 1996] and c a constant. Finally, notice that the work of [Liaskos et al. 2012] disregards any other item attributes apart from the collected statistics p_i, τ_i and the size l_i. Therefore, it allows for the selection of items that may maximize the service ratio, but have no coherence (e.g., semantic) whatsoever. The ensuing scheme will address this shortcoming.

[1] $W(x)$ is the solution to $x = w \cdot e^w$. The condition $W < -1$ is mandatory, according to the analysis of [Liaskos et al. 2012].

3.1. Objective and Workflow

The ideal goal of the present study would be the derivation of an analytical relation of the form:

$$\mathbb{O}(\sigma) = argmax_{(s)}\{SR(Set\, s) |\, Coherence(s) = \sigma\} \tag{6}$$

where σ is a freely defined level of coherence, SR denotes the service ratio of a given item set (e.g. TV shows) and $\mathbb{O}$ is the subset that yields the maximum SR out of all subsets with coherence σ. Coherence is defined as the standard deviation of a given attribute of items in subset s (e.g., the viewer age limit of a subset of movies). This strict definition of optimality is readily NP-Complete to achieve. Enumerating all sets s with coherence σ is an extended boolean satisfiability problem ($k - SAT$), the first known example of NP-Complete problems. Therefore, the study will seek heuristic solutions that offer satisfactory results.

Following the heuristic approach, we are presented with two givens. Firstly, pure coherence-oriented selections are formally tackled by the field of data clustering. Secondly, recent studies have shown that pure service ratio-oriented selections can also be carried out by specially adapted clustering techniques [Liaskos et al. 2012]. Given the wide applicability of clustering techniques in general, the present study will seek to combine coherence and service ratio orientations in one clustering scheme. Since these two orientations are orthogonal, we are presented with the following options:

— Apply 2-stage clustering, first focusing on coherence and then on service ratio (or vice versa). However, as concluded in [Liaskos et al. 2012; 2013], this approach leads to service ratio deterioration. Since the two orientations are orthogonal, the first step has no mechanism to avoid the false classification of items critical to the second stage.
— Apply multi-dimensional clustering, based on a vector comprising the coherence-related and the SR-related attributes. Nonetheless, clustering techniques do not behave well with the increase of clustering dimensions (*curse of dimensionality* [Kriegel et al. 2009]). Furthermore, efficient SR-clustering is non-linear [Liaskos et al. 2012] while coherence-clustering is a linear process, discouraging direct combination at first.

In light of these facts, we introduce biased clustering. A freely varying bias $t \in [0, 1]$ regulates the focus of the overall process. For $t = 0$ the focus is solely on selecting the most coherent subset (e.g., TV shows with similar rating). For $t = 1$ the process will select the item subset that yields the highest service ratio with no coherence concerns. For intermediate values, the process will seek to compromise the two goals. The bias t does not promise linearity. Instead, it is intended as a helping parameter. The expression (6) is transformed as $\mathbb{O}(t)$, $\sigma(t)$. A push scheduling authority is supposed to execute the proposed algorithms for $t \in [0, 1]$, deriving the final $\mathbb{O}(\sigma)$ relation. It can then choose the point of operation, $\mathbb{O}(\sigma_o)$, fitting to its case.

We study this merging of clustering orientations on the context of Centroid-based clustering methods (K-means algorithm [Arthur and Vassilvitskii 2007]) and Density-based methods (DBSCAN algorithm[Ester et al. 1996]).

4. A NOVEL CENTROID-BASED, SERVICE RATIO-ORIENTED CLUSTERING SCHEME

Centroid-based clustering requires a notion of "item distance" and "cluster representatives" (centroids). According to the K-means operation, an item is assigned to the closest centroid, while the centroids themselves are then recalculated as the most representative items of the formed clusters. Let the attributes of an item i be expressed in vector form as follows:

$$\vec{C_i} = \{\ \alpha_{i1}\ \ \alpha_{i2}\ \ \cdots\ \ \alpha_{iA}\ \} \tag{7}$$

where A is the number of attributes. Assume that the α_{ij} attributes refer to semantic or technical characteristics of the items available to the push server at the OLT. Quantifying the similarity of any two items i_1, i_2 in the form of a "distance" Δ is straightforward. Using the Euclidean vector distance produces:

$$\Delta(i_1,\, i_2) = \left|\vec{C_{i_1}} - \vec{C_{i_2}}\right| = \sqrt{\sum_{j=1}^{A} \left(\alpha_{i_1 j} - \alpha_{i_2 j}\right)^2} \tag{8}$$

Picking the most representative item, I (centroid), out of a given set $i = 1 \ldots n$ is also straightforward:

$$I = argmin_{(i)} \left\{ \left| \vec{C_i} - \frac{1}{n} \sum_{i=1}^{n} \vec{C_i} \right| \right\} \tag{9}$$

The expressions (8) and (9) are standard choices for the K-means algorithm. They can be used by the push server for detecting items with similar semantic or technical attributes. However, the service ratio that can be achieved by such a group of items is not taken into account.

4.1. Centroid-based clustering for high service ratio

In order to maintain the facilities of clustering while considering the ratio of successfully served hidden queries we examine expression (4). The expression is a necessary condition for any service ratio-optimal dataset. Failing to comply with (4) results definitely into impaired service ratio. However, we observe that an inequality does not imply crispness. In other words, a set of m items such that:

$$\sum_{i=1}^{m} \frac{\tau_i \cdot l_i}{1 + W\left[\frac{1}{e}\left(\frac{min\left\{\frac{p_i}{\tau_i \cdot l_i}\right\}}{\frac{p_i}{\tau_i \cdot l_i}} - 1\right)\right]} = 10^9 >> -1 \tag{10}$$

may satisfy more queries when pushed to the clients than another set for which:

$$\sum_{i=1}^{m} \frac{\tau_i \cdot l_i}{1 + W\left[\frac{1}{e}\left(\frac{min\left\{\frac{p_i}{\tau_i \cdot l_i}\right\}}{\frac{p_i}{\tau_i \cdot l_i}} - 1\right)\right]} = -0.9999 > -1 \tag{11}$$

This assumption will be used for expressing the query serviceability of two items in the form of a "distance". For ease of expression let x_i denote the ratio:

$$x_i = \frac{min\left\{\frac{p_i}{\tau_i \cdot l_i}\right\}}{\frac{p_i}{\tau_i \cdot l_i}} \tag{12}$$

Equation (4) is transformed as:

$$\sum_{i=1}^{N} \frac{x_i \cdot p_i}{1 + W\left[\frac{1}{e}\left(x_i - 1\right)\right]} > -min\left\{\frac{p_i}{\tau_i \cdot l_i}\right\} \Longrightarrow \left|\sum_{i=1}^{N} \frac{x_i \cdot p_i}{1 + W\left[\frac{1}{e}\left(x_i - 1\right)\right]}\right| < \left|min\left\{\frac{p_i}{\tau_i \cdot l_i}\right\}\right| \tag{13}$$

Consider the case of two data items, $i = i_1, i_2$. Assume further that $x_2 = 1$, i.e. the item i_2 has the lowest $p_i/\tau_i \cdot l_i$ value. The Lambert-W function yields $W[0] \to -\infty$ in the

studied case of $W < -1$. Therefore, the two items must comply with:

$$\left| \frac{x_{i_1} \cdot p_{i_1}}{1 + W\left[\frac{1}{e}\left(x_{i_1} - 1\right)\right]} \right| < min\left\{ \frac{p_i}{\tau_i \cdot l_i} \right\},\ i = i_1,\ i_2 \tag{14}$$

Relation (14) can quantify the push serviceability of any two items. At first, the item with the minimum $p_i/\tau_i \cdot l_i$ ratio is detected. For the remaining item, the quantity $\mathcal{S} = \left| \frac{x_{i_1} \cdot p_{i_2}}{1+W\left[\frac{1}{e}\left(x_{i_1}-1\right)\right]} \right|$ is calculated. If $\mathcal{S}$ is smaller than $\mathcal{R} = min\left\{p_i/\tau_i \cdot l_i\right\}$, the items are compatible and may be broadcasted together in a way that maximizes the client service ratio. If not, the items are incompatible. Therefore, we can define the new distance metric as follows.

Definition 4.1. The distance, $\mathcal{D}$, quantifies the serviceability of two data items, $i = i_1$, $i = i_2$ in a push system and is calculated as:

$$\mathcal{D} = \begin{cases} \frac{S}{\mathcal{R}-S} & , 0 \leq \mathcal{S} < \mathcal{R} \\ \infty & , \mathcal{S} \geq \mathcal{R} \end{cases} \tag{15}$$

where

$$\mathcal{S} = \left| \frac{x_{i_1} \cdot p_{i_1}}{1 + W\left[\frac{1}{e}\left(x_{i_1} - 1\right)\right]} \right|, \quad \mathcal{R} = min\left\{ \frac{p_{i_1}}{\tau_{i_1} \cdot l_{i_1}}, \frac{p_{i_2}}{\tau_{i_2} \cdot l_{i_2}} \right\} \tag{16}$$

The serviceability distance $\mathcal{D}$ will be denoted as SERV.

SERV can be perceived as a non-Euclidean distance between two data items. Notice that the calculations of $\mathcal{R}$ and $\mathcal{S}$ do not involve other quantities but the collected item statistics $\left\{p_i,\, \tau_i\right\},\ i = i_1, i_2$ and their sizes $l_i,\ i = i_1, i_2$. It is the counterpart of the Euclidean distance, Δ, of equation (8) that refers to the similarity over semantic or technical attributes, α_{ij}, of the items.

Apart from a metric of distance, centroid-based clustering requires a way of electing a cluster representative. In the case of the Euclidean distance, Δ, this is accomplished by averaging the attribute vectors of the items (eq. (9)). As SERV is non-linear, averaging is not applicable. However, notice from the $\mathcal{R}$ quantity of equation (16) that the item with the smallest $p_i/\tau_i \cdot l_i$ ratio defines the form of the SERV distance, $\mathcal{D}$. This observation can be used for defining cluster representatives when the SERV distance metric is used instead of the Euclidean Δ.

Remark 4.2. When SERV is used as the distance metric of centroid-based clustering, the cluster representatives, I^*, are defined as:

$$I^* = argmin_{(i)} \left\{ \frac{p_i}{\tau_i \cdot l_i} \right\} \tag{17}$$

where i belongs to the index set of the given cluster. This choice comes naturally, considering that items with smaller popularity but greater size and criticality stand out as non-profitable choices for data push. This becomes apparent if we visualize a 10-hour long movie that is awaited by only a trivial percentage of the viewers. These shows should quickly form clusters of their own (along any other similar ones), discouraging their participation to other selections.

As an example of practical operation, the K-means algorithm can use equations (15) and (17) as follows. Given N shows, the IPTV service at the OLT selects some random centroids. The number of centroids is user-defined. Every show $i = 1 \ldots N$ is assigned to the nearest centroid based on the SERV distance, $\mathcal{D}$. New centroids are calculated

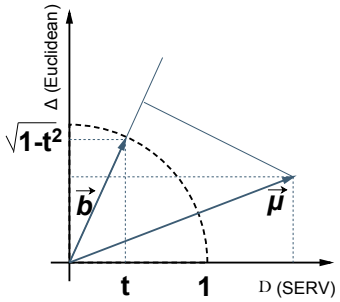


Fig. 2: Combining the SERV (service ratio) and Δ (coherence) metrics with the use of a user-supplied bias vector. Selecting a value of $t \in [0, 1]$ results into a balanced serviceability and coherence orientation.

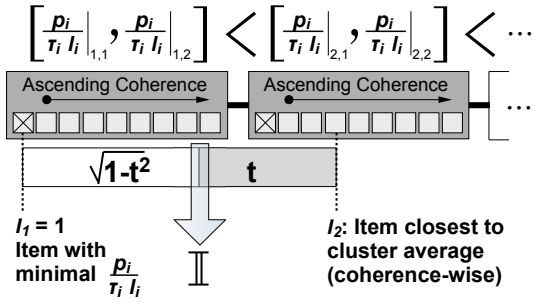


Fig. 3: Visualization of the novel, tunable centroid update scheme. The new cluster centroid, $\mathbb{I}$, is selected proportionally to the tuning factor t in the range defined by the SERV and Euclidean update schemes.

for each cluster next, using equation (17). The process is repeated iteratively until convergence (no new item assignment) or expiry of a computational time limit. The push server can finally select and use an appropriate cluster of shows out of the formed ones.

Still, the SERV distance $\mathcal{D}$ and the Euclidean Δ are not directly related. Furthermore, their goals can be in direct opposition. For example, the most promising subset of IPTV shows in terms of service ratio can have little thematic coherence. The next step is to introduce a tunable balance between the two approaches.

4.2. Tuning serviceability and coherence of pushed data

Assume two data items, i_1 and i_2. Using equation (8) the $\Delta(i_1, i_2)$ distance is produced. From equation (17) the SERV distance $\mathcal{D}(i_1, i_2)$ is derived. Notice that the two metrics cannot be combined in a straightforward manner. They refer to different item attributes and use different calculation processes (linear and non-linear). In order to introduce a sense of tunability, the two metrics are placed on the two two-dimensional system of axes in Fig. 2. Thus, the vector $\vec{\mu}$ is formed as:

$$\vec{\mu} = \{\, \mathcal{D},\ \Delta \,\}. \tag{18}$$

Furthermore, we employ the standard unary vector:

$$\vec{b} = \left\{\, t,\ \sqrt{1-t^2} \,\right\},\ t \in [0, 1] \tag{19}$$

The tuning parameter t is supplied as an input and regulates the rotation of the $\vec{b}$ vector. For $t = 0$, $\vec{b}$ is parallel to the Δ axis, while for $t = 1$ it becomes parallel to the $\mathcal{D}$ axis. The vector $\vec{b}$ expresses the bias of the clustering process, tuning it between full coherence and full serviceability orientation. The combination of Δ and $\mathcal{D}$ is the projection of $\vec{\mu}$ on $\vec{b}$.

Definition 4.3. The tunable distance $\mathbb{D}$ that balances the serviceability and coherence of two data items is defined as:

$$\mathbb{D}(t) \ = \ \vec{\mu} \cdot \vec{b} = t \cdot \mathcal{D} + \sqrt{1-t^2} \cdot \Delta \tag{20}$$

where $t \in [0, 1]$ is the supplied tuning factor.

From another aspect, the angles defined by the bias vector, $\vec{b}$, and the $\mathcal{D}$-, Δ-axes act as weights on the SERV and Euclidean distance metrics, tuning the significance of $\mathcal{D}$ over Δ and vice-versa. In general, a weight function projects a vector of measurements, $\vec{\mu}$, over a given direction $\vec{b}$, which defines the significance of each of the vector's dimensions [Grossman et al. 1980]. In this sense, other alternatives can also be used instead of equation (19). We prefer the unary vector $\vec{b}$, defined in equation (19), because it ensures that all possible bias vectors are of equal significance, in the sense that it holds $\left\|\vec{b}\right\| = 1$ for all t values.

As a final step we proceed to combine the centroid update methods of expressions (9-Euclidean) and (17-SERV). The Euclidean centroid update method picks the item that is most similar to the cluster average. On the other hand, SERV picks the item with the smallest $\{{}^{p_i}/_{\tau_i \cdot l_i}\}$ value. In order to define a tunable combination of the two mechanisms we work as follows. At first we note that push systems typically treat items in groups (called "buckets"), in order to limit the complexity of the scheduling process [Vaidya and Hameed 1999]. As a rule of a thumb, 10 virtual "buckets" are usually sufficient for near-optimal operation [Vaidya and Hameed 1999]. Thus, the range $[\min\{{}^{p_i}/_{\tau_i \cdot l_i}\}, \max\{{}^{p_i}/_{\tau_i \cdot l_i}\}]$ is split into equal sub-ranges, resulting into a corresponding categorization of the items as well[2]. Items within each sub-range are sorted by descending Euclidean distance from the Cluster Average (i.e., ascending coherence, Fig. 3). As an exception, the item with minimal ${}^{p_i}/_{\tau_i \cdot l_i}$ value is placed first. Thus, a strict-weak ordering of the items is attained, as shown in Fig. 3.

In this sorting, the item with index $I_1 = 1$ represents the centroid choice of the SERV clustering. The index I_2 is the choice of the Euclidean centroid update process and is bound to be in the new sorting. Employing the same tuning factor t as in Definition 4.3 produces:

$$\mathbb{I}(t) = \left[\left[t \cdot I_1 + \sqrt{1-t^2} \cdot I_2\right]\right] = \left[\left[t + \sqrt{1-t^2} \cdot I_2\right]\right] \tag{21}$$

where $[[.]]$ is the rounding function. Once again, for $t = 0$ the centroid update process, $\mathbb{I}$, coincides with the Euclidean selection. For $t = 1$ it coincides with the SERV selection scheme. Thus, the tunable distance $\mathbb{D}(t)$ and the tunable centroid selection $\mathbb{I}(t)$ are inline. Qualitatively, as t increases, $\mathbb{I}(t)$ moves towards items with lower index than I_2. While the new centroid is within the same sub-range as I_2, this movement results into decreasing coherence, while serviceability remains approximately constant. Increasing t further moves the new centroid into a new sub-range which offers better service ratio. Coherence presents a discontinuity while traversing the two adjacent sub-ranges, and begins to decrease again. Thus, serviceability is an increasing function of t, while coherence is a piece-wise decreasing function of t. Discontinuity is natural, given that serviceability and coherence are unrelated system aspects in the general case. We also note that it is possible for a sub-range to contain completely incoherent items. In this case, t can be increased further, in order to skip the sub-range. In the case where all sub-ranges are incoherent, the proposed process may not produce satisfactory results. On the other hand, balancing coherency and serviceability may not be meaningful altogether in such cases.

Remark 4.4. The described centroid update process prioritizes service ratio over coherency due to the weak ordering by $\{{}^{p_i}/_{\tau_i \cdot l_i}\}$. This choice is due to the fact that the original expression of the optimization goal (6) targets optimal service ratio for a given coherence level. Furthermore, it is known that item selection for optimal service ratio

[2]Unequal splitting can also be employed, in order to ensure that each sub-range contains a minimum number of items [Varga and Fakhamzadeh 1997].

Algorithm 1 A K-means variation for tuning data selection between push serviceability and data coherence.

INPUTS:
❶ A set of $i = 1 \dots N$ items, with statistics $\{p_i, \tau_i\}$, sizes l_i and coherence attributes $\{\alpha_{ij}, j = 1 \dots A\}$.
❷ A tuning factor value $t \in [0, 1]$.
❸ A maximum number of potential clusters, C_{max}.
❹ The number of repetitions, R_1, R_2.
OUTPUT: The subset balancing maximum service ratio and data coherence, $best\_subset(t)$.

1: Init $best\_subset = []$;
2: Init $Best\_Coverage = 0$;
3: **FOR** $repetition = 1 : R_1$
4: **FOR** $C = 1 : C_{max}$
5: Select C random items as centroids;
6: Set $r = 1$;
7: **WHILE** (Item assignment change detected) OR ($r \geq R_2$)
8: $r = r + 1$;
9: Assign each item i to nearest centroid by $\mathbb{D}(t)$;
10: Calculate new centroids, C^* by $\mathbb{I}(t)$;
11: C=C^*;
12: **ENDWHILE**
13: **FOREACH** formed cluster CL
14: **IF** $\sum_{i \in CL} p_i > Best\_Coverage$
15: $Best\_Coverage = \sum_{i \in CL} p_i$;
16: $best\_subset = CL$;
17: **ENDIF**
18: **ENDFOREACH**
19: **ENDFOR**
20: **ENDFOR**

is a very sensitive process [Liaskos et al. 2012; 2013]. A few inappropriate item selections can impair the service ratio considerably. Thus, the prioritization of the $\{p_i/\tau_i \cdot l_i\}$ ordering comes as a natural choice. Furthermore, clustering processes are hill-climbing algorithms that tend to converge to local optima in general. The described, shuffled ordering from the K-means aspect can act as a controlled kick-away, encouraging the exploration of further solutions that are potentially nearer to the global optimum.

5. APPLICATIONS

5.1. Tunable clustering with the K-means algorithm

Algorithm 1 constitutes a showcase of how the tunable distance $\mathbb{D}$ and the centroid selection $\mathbb{I}$ can be incorporated to the K-means algorithm. The algorithm uses the K-means functionality and its core process takes place in lines $5 - 12$. Given N data items to be partitioned into C subsets, the algorithm first selects an equal number of centroids at random. This probabilistic selection means that the algorithm must be repeated several times in order to avoid being trapped at a local optimum. This is achieved by the outermost repetition loop at line 3. Furthermore, K-means requires the number of clusters C to be supplied as an input. Since there is no way of detecting the optimal number of clusters in advance, the algorithm performs several executions with different C input. This is expressed by the for loop at line 4. Once the random, initial

centroids have been selected, the algorithm assigns each other item to the nearest centroid (line 9). Thus, C clusters are formed. New centroids are calculated for these new clusters and the process is repeated in the while-loop until the item assignments remain unaltered in subsequent iterations. At this point the while-loop has converged to a final clustering, CL. The cluster of CL that provides the maximum coverage is selected over all repetitions and C values. The logic behind this choice is that a subset that provides, e.g., $\sum p_i = 90\%$ coverage will lead to a higher service ratio than another that corresponds to $\sum p_i = 1\%$.

Finally, Algorithm 1 returns this $best\_subset$ that balances the maximum service ratio and data coherence orientations. Notice that this subset may not necessarily uphold condition (4). In this case, the items with the highest distance, $\mathbb{D}$, from the centroid of the $best\_subset$, $\mathbb{I}$, are discarded iteratively from the cluster until the condition holds. Other approaches are also possible as follows. If the service ratio is prioritized, the item with the smallest $p_i/l_i \cdot \tau_i$ ratio is discarded iteratively from the cluster. A small $p_i/l_i \cdot \tau_i$ ratio implies low coverage (popularity p_i) but large size (l_i) and high criticality (τ_i) and, therefore, the corresponding item is too demanding for its service ratio profits. Finally, if coherence is prioritized, we discard the items with the highest Euclidean distance Δ from the coherence-centroid of the set, which is the vector:

$$\vec{a} = E_i \left[a_{ij} \right], \forall i \in best\_subset\,, \tag{22}$$

defined as average over all rows of a_{ij} that correspond to items within the $best\_subset$.

Algorithm 1 has the same complexity as K-means, given that the latter constitutes the core of the Algorithm (lines $5-12$). The original K-means has a computational complexity of $O(c \cdot n \cdot a \cdot r)$, where n is the number of items to be clustered, c is the required number of clusters, a is number of attributes per item and r is the maximum number of allowed iterations for the inner *while* loop (line 7) [Arthur and Vassilvitskii 2007]. Additionally, Algorithm 1 executes K-means R_1 times with random seed, for all values of $c = 1 \ldots C_{max}$. Furthermore, it considers A coherence attributes and one service ratio-related attribute ($p_i/l_i \cdot \tau_i$) per item. Thus, the computational complexity of Algorithm 1 is $O\left(\frac{C_{max}(C_{max}+1)}{2} \cdot R_1 \cdot R_2 \cdot N \cdot (A+1)\right)$. In terms of memory footprint, Algorithm 1 requires $3 + A$, N-sized arrays to hold the p_i, l_i, τ_i and a_{ij} input parameters. An additional N-sized array is needed per item, in order to designate its nearest centroid. The centroids themselves are stored in a C_{max}-sized array, noting that $C_{max} \leq N$. Similarly, an array with maximum size N is need for holding the best cluster, CL. Thus the total memory overhead is $O\left((6 + A) \cdot N\right)$.

5.2. Employing SERV in density-based clustering

The ability to tune serviceability and coherence can be easily incorporated to density-based clustering as well.

Density-based clustering is preferred when the dataset representation exhibits clearly separated dense and sparse areas. The algorithms of this family are able to detect the optimal number of clusters, but require extra inputs in its place. An item belongs to a cluster if it is within ϵ distance units away from K members of the cluster. The way of operation comprises three steps:

— Select a previously unvisited item and check for other items within ϵ. If their number is more than $K - 1$, form a cluster. Else, mark as visited and discard the point.
— Repeat the process for items within the cluster until no new additions are found.
— Repeat from the first step until all available items have been visited.

A density-based clustering algorithm receives the distance metric type, K, and ϵ as inputs and returns data clusters as output. For example, the DBSCAN algorithm can be expressed in pseudocode as:

$$Clusters = DBSCAN(DATA,\, Distance\,Metric,\, \epsilon,\, K) \tag{23}$$

Thus, imbuing DBSCAN with push serviceability considerations is straightforward. The tunable distance $\mathbb{D}(t)$ can serve as the $Distance\,Metric$ without changes. K is a design choice (how many items to be pushed at a minimum) and is irrelevant to the tuning operation. The ϵ parameter is relevant only when coherence is the sole clustering criterion (i.e., $t = 0$). In this case, ϵ can be defined in any appropriate way. For example, [Daszykowski et al. 2001] defines a generic approximation of the parameter as:

$$\epsilon_0 = \left(\frac{\left(\prod_j \left(\max_i \{\alpha_{ij}\} - \min_i \{\alpha_{ij}\} \right) \right) \cdot K \cdot \Gamma\left(1 + \frac{A}{2}\right)}{N \cdot \sqrt{\pi^A}} \right)^{\frac{1}{A}} \tag{24}$$

When $t = 1$, the α_{ij} coherence attributes become irrelevant, as the focus is solely on the collected statistics, p_i, τ_i, and on the unconditional maximization of the service ratio. Thus, ϵ can take any large value, e.g.,:

$$\epsilon_1 = max\{\mathcal{D}_{ij} | over\, all\, i, j\} \tag{25}$$

For all intermediate values of the tuning factor t, $\epsilon(t)$ can be set as:

$$\epsilon(t) = \epsilon_1 \cdot t + \epsilon_0 \cdot \sqrt{1 - t^2} \tag{26}$$

which leads to the novel, tunable $DBSCAN_T$ variation:

$$Clusters = DBSCAN_T(DATA,\, \mathbb{D}(t),\, \epsilon(t),\, K) \tag{27}$$

The push server then simply selects the resulting cluster with the greatest coverage, as in lines 11-16 of Algorithm 1. Again, if condition (4) does not hold for the chosen subset, items can be discarded iteratively as discussed in Section 5.1.

5.3. Additional Convergence and Complexity considerations

The SERV distance and the centroid update procedure are non-linear. In that sense, the stability and convergence of the proposed clustering schemes needs to be examined further. We will show that clustering in general can be classified as a scheme of Alternating Optimization (AO) [Bezdek and Hathaway 2003]. The later has been extensively studied lately in terms on global and local convergence criteria.

CLAIM 1. *A clustering scheme is a sub-case of Alternating Optimization.*

Proof. Assume a generic process $f(X_1, X_2, \ldots, X_k)$. AO is an iterative procedure that seeks to optimize f by alternating individual optimizations over the inputs $X_1, X_2, \ldots, X_k$. Let X_j represent the subset of items s_j out of a given set $\mathbf{s}$ that belong to an arbitrary group j defined by a participation criterion. Clustering is defined as an iterative process of sequential assignment of items to the groups $j = 1 \ldots k$, terminated when the standard deviation of items in each group is minimized, QED. ■

According to [Bezdek and Hathaway 2003; 2002] the convergence rate of AO is linear, regardless of the norm (in our case linear or non-linear) used for measuring the distance between two solutions to the optimization problem. The only prerequisite is

that the norm be equipped with comparison operators ($>, <, =$). Notice from equation (15) that the SERV norm is scalar and comparable by standard operators, despite its non-linearity. We therefore conclude that:

LEMMA 5.1. *The convergence rate of a clustering process classified as AO remains unaltered by the incorporation of the SERV metric.*

Complexity-wise, incorporating SERV distances and update mechanisms to either K-means or DBSCAN requires a trivial computational overhead. Furthermore, clustering is typically an offline and heavily parallelizable process. The complexity of $DBSCAN_T$ is equal to that of the original $DBSCAN$, i.e. $O(N \cdot logN)$ [Birant and Kut 2007; Ester et al. 1996]. The non-linear centroid update process of K-means is approximately $O(N \cdot logN)$ as well, due to the sorting by ascending $p_i/l_i \cdot \tau_i$, $i = 1 \dots N$ ratio.

6. SIMULATIONS

This Section evaluates the analytic conclusions in the context of a simulated push system in the MATLAB environment [MathWorks-Inc 2010]. The goals of the evaluation are the following:

— The analysis introduced a new, non-linear distance metric for serviceability quantification (SERV). Thus, we first demonstrate that SERV-based clustering achieves higher service ratio than the alternatives.
— Tunability of data coherence and serviceability is a core-point of the analysis. Simulations evaluate this aspect next.

Both goals are tested extensively through Monte Carlo runs, for centroid and density-based clustering.

The benchmark considers related solutions from the field of push-scheduling [Jiang and Vaidya 1999; Raissi-Dehkordi and Baras 2007], as well as well-known data clustering algorithms (K-Means, DBSCAN). These approaches are complimentary in their nature. On one hand, the selected push-scheduling solutions focus exclusively on the maximization of the client service ratio, without data coherence concerns. The *Service Scheduler* does so by minimizing the mean query service time [Jiang and Vaidya 1999]. The *Impatience Scheduler* minimizes the clients' *impatience*, a metric defined as an exponential function of the service time [Raissi-Dehkordi and Baras 2007]. On the other hand, clustering algorithms target data coherence, without client service ratio concerns.

6.1. Setup

The proposed algorithms are applicable to any push system. However, in order to present an example of practical application, we consider an implementation over the PON architecture of Fig. 1. There exist 10 ONUs connect to $1,000$ clients each. The fiber channel delay is considered negligible for all clients. Since the user statistics are considered to be known, the upstream functionality is not a part of the simulations.

We once again consider the interactive IPTV example. The OLT hosts N multimedia items (shows), their number and statistics varying per study. The τ_i criticality parameters are picked at random from the range $(0, 1)$ uniformly. The p_i popularity statistics are set as a random permutation of the Zipf distribution in each run [Pietronero et al. 2001], i.e.:

$$p_{1\dots N} = RandPerm\left[\frac{i^{-\theta}}{\sum_{i=1}^{N} i^{-\theta}}, i = 1 \dots N\right], \theta > 0 \tag{28}$$

This choice allows for controlling the presence of highly popular shows (high θ values) while retaining the probabilistic nature of the assignment. Additionally, we notice that the service ratio of a show is affected by its ${}^{p_i}/_{\tau_i \cdot l_i}$ values. Since p_i and τ_i are already varied randomly, the variation of l_i does not contribute to the experiments further. Thus, the sizes l_i are set to one unit ($1\ G$Byte) for every show. The coherence attributes, α_{ij}, of the shows are derived from real or synthetic datasets, as described per case.

The OLT selects a subset of the N available shows and pushes it periodically to the clients. The metric of their coherence is the quantity:

$$\sigma \left[\left| \vec{C_i} - \frac{1}{n} \sum_{i=1}^{n} \vec{C_i} \right|, i \in Subset \right] \tag{29}$$

where σ expresses the standard deviation and $\vec{C_i}$ is the vector of item attributes from expression (7).

The push process takes place as follows. At the initialization stage, the occurrence ratios (v_i) are set for each show in the chosen subset. These are derived from equation (5). The operation stage is a continuous cycle, selecting the next show to be pushed when the transmission of the current one has finished. Preemption is not used. In accordance with related studies [Vaidya and Hameed 1999], the selection process picks the show whose last occurrence time deviates the most from its optimal push period of equation (3). In other words, if t_i^* is the time when item i was last seen and t is the present time, then the next item, ι, to be broadcast is selected as $\iota = argmax_{(i)} \{|(t - t_i^*) - w_i|\}$.

The clients connected to the OLTs may have one hidden query pending at any given time. A random query interarrival of up to $10sec$ (uniform) ensures this condition. This model essentially corresponds to a single-threaded client. Grouping several threads to model more complex clients is straightforward and beyond our scope. Each query is drawn from the show popularity distribution, p_i. Once the needed show index i is thusly defined, the maximum waiting time (in sec) of the corresponding client is drawn from the exponential distribution of equation (1). If this deadline expires and the corresponding show has not started, the query is considered unsuccessful and is discarded.

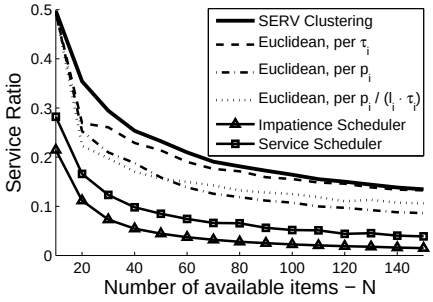
Each simulation assumes a warmup period lasting for $10,000$ dispatched queries (answered or not). Then, the hidden query service ratio is measured over the next $300,000$ queries. The confidence interval achieved is typically at the level of 95%.

Finally, a Monte Carlo experiment comprises 1000 such runs with random seeds. The outcome of a Monte Carlo experiment is the mean service ratio and the mean standard deviation (eq. (29)) over all 1000 repetitions.

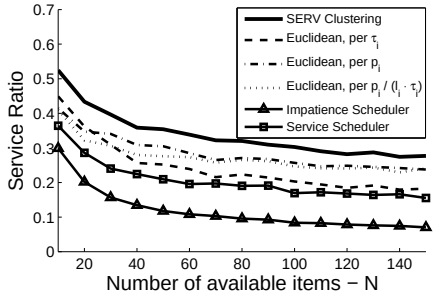
6.2. Evaluation of tunable centroid-based clustering (K-means).

Using the novel SERV distance and associated centroid update process, the K-means algorithm receives a considerable boost in terms of service ratio, as shown in Fig. 4. The tuning factor t is set to zero, in order to study the service ratio potential of the algorithms. The SERV-based K-means is shown to yield higher service ratio than:

— The euclidean distance variations of K-means, consisting of clustering the same items by popularity, criticality or ${}^{p_i}/_{l_i \cdot \tau_i}$ ratio using the standard, Euclidean distance and centroid update mechanism.
— The related approaches, [Jiang and Vaidya 1999] (*Service Scheduler*) and [Raissi-Dehkordi and Baras 2007] (*Impatience Scheduler*).



(a) Comparison of service ratios achieved by the proposed scheme (SERV) and five alternatives. The plot corresponds to popularity distributions with limited skewness ($\theta = 0.3$). The SERV-based Clustering outperforms all related solutions in every examined case.



(b) Achieved service ratios in the case of highly skewed popularity distributions ($\theta = 0.9$). The advantage of SERV over the alternatives is retained in all cases.

Fig. 4: Comparison of the SERV-enabled K-means with alternative heuristics. The addition of the SERV distance metric to K-means makes it the best performing algorithm for service ratio-optimal item selection.

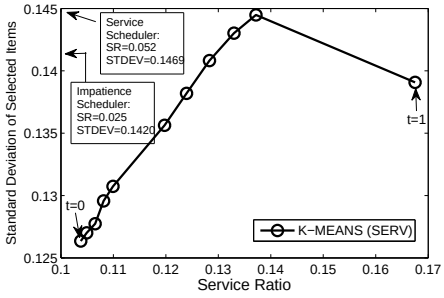


Fig. 5: Service ratio versus Coherence plot for the SERV-enabled K-means. Popularity distribution with limited skewness is assumed ($\theta = 0.3$). The circles represent incremental steps in the t parameter by 0.1. The graph is almost linear as t increases from 0 to 0.9. Service ratio optimization ($t = 1$) typically leads to very small subsets (e.g. containing $1 - 10\%$ of the available N items). Thus, coherence increases in the final step.

The x-axis represents the number of available data items. A separate Monte Carlo experiment is executed for each value of $N = 10 : 10 : 150$. The proposed scheme clearly outperforms the alternatives in all cases.

The θ parameter of the Zipf distribution is set to 0.3 in Fig. 4a and to 0.9 in Fig. 4b. The value $\theta = 0.3$ is rather small, representing a case where the dataset does not contain items with outstanding popularity. In other words, there are no "easy" choices for efficient data push and the service ratio margin is expected to be limited. This is furtherly aggravated as N increases, leading to the decreasing form of the plots. Even in this case of limited serviceability prospects, the SERV-enabled K-means outperforms the alternatives. Even at the highest N values, SERV retains an advantage over the compared approaches. In Fig. 4b, where θ is raised to 0.9, the gap between SERV and the alternatives increases. In this case there exist items with exceptionally high cov-

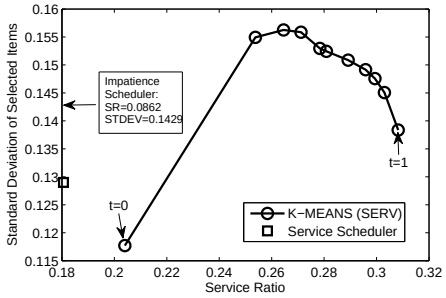


Fig. 6: Service ratio versus Coherence when $\theta = 0.9$ (highly skewed popularity distribution). The circles represent incremental steps in the t parameter by 0.1. The graph retains the same form as in Fig. 5. The large transition from $t = 0$ to $t = 0.1$ is due to the presence of an item with high $p_i/\tau_i \cdot l_i$ ratio.

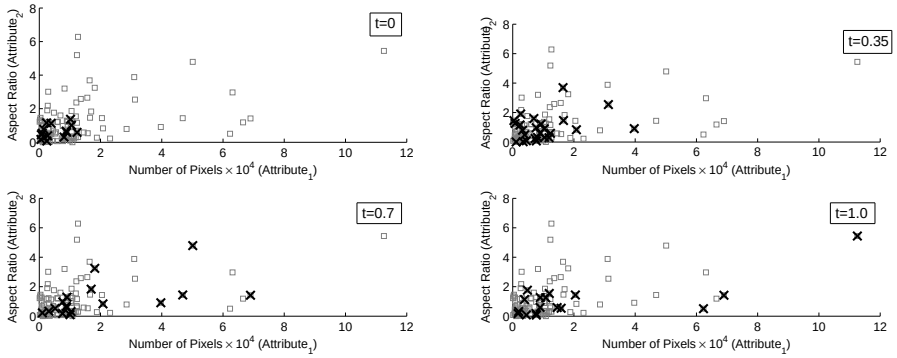


Fig. 7: The effects of the tuning factor in the data subset selection. When $t = 0$ the algorithm will simply select the most coherent data (upper plot). As t increases, coherence is gradually disregarded in favor of the service ratio. Thus, the data selection now covers the complete set (lower plot).

erage which constitute easy choices. However, SERV detects even more fitting subsets in all cases. The low performance of *Service* and *Impatience* schedulers also highlights the importance of careful data selection. Despite their exclusive orientation towards maximum service ratio, these approaches employ the complete set of data items. However, this increases the total load of the communication channel. Thus, the average push period per item increases as well, resulting into query timeouts and low service rate.

Having demonstrated the efficiency of SERV and the associated centroid update process in terms of service ratio, we proceed to study its tunability in terms of data coherence.

The tunability of the SERV-enabled K-means is examined in Fig. 5 and 6 which consider $N = 100$ data items and $\theta = 0.3$, 0.9 respectively. There exist two coherence attributes per item $\{\alpha_{i1}, \alpha_{i2}\}$ which are picked at random in $(0, 1]$ (uniform) at every run. The tuning factor t varies in $0 : 0.1 : 1$. We consider that service ratio is prioritized if Algorithm 1 needs to discard items from the $best\_subset$. For each value of t, a Monte Carlo experiment is executed, logging the mean service ratio and standard deviation σ of the selected subset, forming the two plots.

Starting from $t = 0$, the proposed scheme achieves almost linear tuning of service ratio versus coherency. As t increases, the standard deviation of the selected subset increases while coherence drops. Simultaneously, the service ratio increases considerably. This steady increase continues up to a point ($t = 0.9$ in Fig. 5 and $t = 0.3$ in Fig. 6). From that point and on, both the service ratio and the data coherency benefit. The reason for this behavior is the high selectivity of the service ratio optimization process. As an example, consider that the total set of shows (items) contains a movie of very high $p_i/\tau_i \cdot l_i$ ratio. This could be due to high popularity, low criticality and size, or both. A purely service ratio-optimal selection may include just this one item. The coherence of the selected subset will be maximized, since it contains just one show. This extreme example shows that the concave form of Fig. 5 and 6 is naturally expected in the general case.

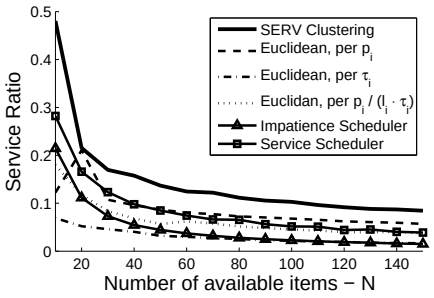
We also notice that both Figures exhibit one sharp transition with regard to the tuning parameter t. The transition is much greater in Fig. 6 which refers to random popularity distributions with very high skewness. To explain the phenomenon we calculate the minimum and maximum item popularity in the two cases examined by the Figures. In Fig. 5, the θ parameter of equation (28) is set to 0.3. This translates to a maximum popularity of 0.023 and a minimum of 0.007. In Fig. 6, this difference in much wider: 0.157 versus 0.002. It is apparent that both cases contain items with increased $p_i/\tau_i \cdot l_i$ values. It is therefore natural for the two Figures to contain sharp transitions when the selection process affects these items. The transition becomes much sharper as the skewness of the distribution increases.

Finally, Fig. 7 illustrates the effects of the tuning factor on the data subset selection. We assume that the items have two coherence attributes derived from the real data set of [Etzioni et al. 1999; University of California Irvine, School of Information and Computer Sciences 1998]. The data set describes a collection of web images (advertisements) categorized by: i) their size (number of pixels), and ii) their aspect ratio. The image size and aspect ratio are mapped to the coherence attributes $\{\alpha_{ij}, j = 1 \dots 2\}$ of Algorithm 1. In this case, the selection process derives a set of advertisements that maximizes the number of attracted clients, under the condition that all selected images have similar size. These attributes are placed on the axes of Fig. 7, where the gray squares represent the data items. At first, we require the selection of the most coherent subset, i.e., a collection of images with the most similar attributes. This is expressed by a tuning factor of $t = 0$ at the upper plot. The selected items (denoted by the 'x' markers) are near the axes origin as expected. When a maximum service ratio is required ($t = 1$, lower plot) the selection covers the complete data set, disregarding coherence.

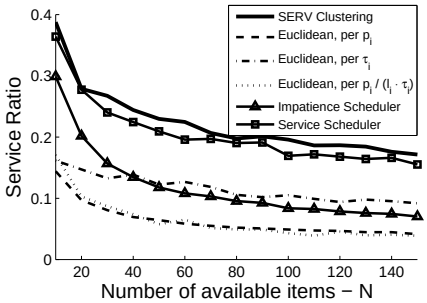
6.3. Evaluation of tunable density-based clustering ($DBSCAN_T$).

The experiments of Fig. 4-6 are repeated for the $DBSCAN_T$ algorithm of Section 5.2. As in Fig. 4, the goal is to achieve the maximum push service ratio, without coherence concerns, in order to demonstrate the efficiency of the SERV distance (i.e., $t = 0$). The results are once again in clear favor of the proposed scheme. Despite the probabilistic nature of the runs (each N value is a separate Monte Carlo run), $DBSCAN_T$ performed much more efficiently than the alternatives in every single case. Notice that the benefits of SERV are greater than in the case of K-Means. This remark holds for any popularity distribution, either less skewed (Fig. 8a, $\theta = 0.3$) or highly skewed (Fig. 8b, $\theta = 0.9$).

The coherence-serviceability tunability tests (Fig. 9) produce interesting results. The coherence attributes are derived from an online database of synthetic data [University of Eastern Finland, School of Computing 2012]. Particularly, we employ the $S1$ dataset, which contains synthetic 2-D data, organized into well-defined Gaussian clus-



(a) Achieved service ratios for popularity distributions with limited skewness ($\theta = 0.3$). The proposed $DBSCAN_T$ outperforms the alternatives considerably in all cases.



(b) The performance of $DBSCAN_T$ is retained for highly skewed popularity distributions ($\theta = 0.9$).

Fig. 8: Service ratio benefits of the use of SERV in DBSCAN ($DBSCAN_T$).

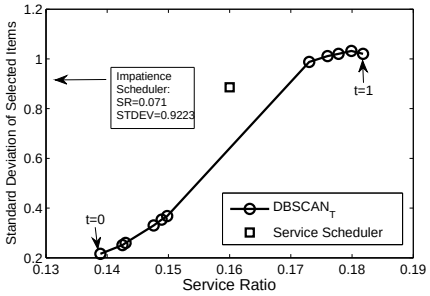


Fig. 9: Service ratio versus Coherence plot for the proposed $DBSCAN_T$ and its alternatives. The circles represent incremental steps in the t parameter by 0.1. $DBSCAN_T$ exhibits almost linear tunability. The sharp transition is due to the density-oriented data sets (e.g. Fig. 10).

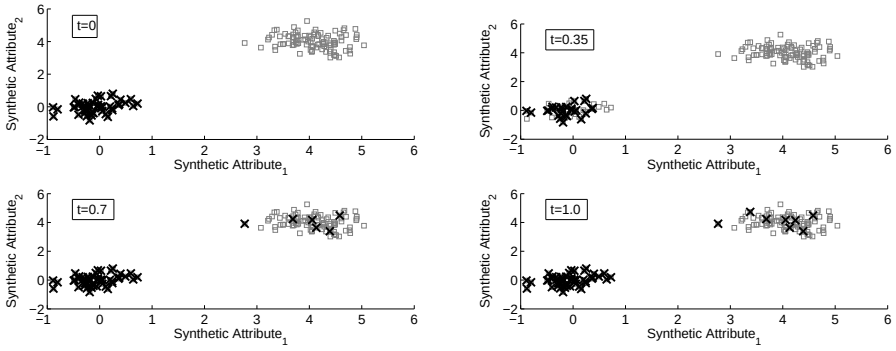


Fig. 10: As the t factor increases from 0 (full coherence) to 1 (full serviceability), the data selection progressively expands to distant clusters. This phenomenon leads to the sharp transition in the plot of Fig. 9.

ters with small overlaps. We choose the first 2 clusters from the $S1$ dataset subset and for each item i we map its coherence attributes, $\{\alpha_{ij},\, j = 1,\, 2\}$, to a random cluster point.

The related approaches are outperformed in both service ratio and coherency. However, notice that the plot exhibits a linear transition rather than the concave form of Fig. 5 and 6. The explanation for this phenomenon becomes apparent when we observe the form of the dataset (Fig. 10). The axes are labeled as $attribute_{1,2}$ since they carry no physical meaning. When $t = 0$, $DBSCAN_T$ will simply select the most coherent subset (classic clustering behavior). However, as t increases the choice will expand to other clusters as well. At that point the average coherence of the subset will decrease sharply. On the other hand, this expansion will give access to more profitable items in terms of serviceability. At $t = 1$ the subset selection has expanded to the complete data set (lower part of Fig. 10). A slight concavity is still observed in the rightmost part of the Fig. 9, between $t = 0.8$ and $t = 1.0$. However, the sharp transition in terms of coherence overshadows the phenomenon.

6.4. Discussion

The simulation results demonstrated the efficiency of the SERV metric in both centroid and density-based clustering algorithms. The behavior of both K-means and DBSCAN was shown to improve greatly when searching for data with high push serviceability. Finally, the ability to balance serviceability and data coherence in a tunable manner is retained in both cases, despite the different way of operation of the algorithms and the much different form of the datasets.

The proposed, coherence-aware, service ratio-optimal data selection scheme can be applied to any push system in general. Regarding the example of IPTV over PON adopted throughout the study, the proposed scheme provides the advantage of automatic selection of shows from immense pools such as the Web. An IPTV service can simply create the Coherency Vs Service Ratio graphs corresponding to its pool and narrow down on a selection that: i) fits in its thematic scope (e.g., education and entertainment), and ii) increase the numbers of its perspective subscribers. An IPTV service may also make controllable deviations from its standard theme (e.g., the addition of short news briefings) when the selection scheme foresees a considerable profit in terms of service ratio. Most notably, the study shows that there exist cases where a service can increase both its service ratio and thematic coherence (Fig. 5 and 6). The proposed solution successfully detects and exploits these opportunities.

While IPTV is intended as a widely accessible example, the applicability of the proposed scheme is extended to any kind of system relying on data-push delivery. Web firms such as Google employ HTTP server push (part of the HTML5 specification) to push advertisements to users globally. The exact push time-schedule is also derived on statistics referring to subject popularity (user views/clicks) and criticality (time spent on a web site). The selection scheme proposed in this study can then extract selections of advertisements that not only fit a given theme, but also promise a high market penetration rate. Notice that HTTP server push is a generic push technology, exemplary used to implement the Apple Push Notification Service, the Android Cloud to Device Messaging Service, the Reverse Ajax technology and the Flash XMLSocket relays.

Finally, in the context of passive optical networks, the proposed scheme also acts as a downstream traffic scheduler. The selected data items are forwarded in priority mode through downstream channels with assured bandwidth. Should the forwarding of the discarded items be imperative, they can be disseminated through channels of non-assured or best effort bandwidth.

7. CONCLUSION

The advent of cost-effective passive optical networks will extend the coverage of push advertising services to extremely wide markets. Selecting which data to push affects the service ratio of the system. The present work proposed a novel data selection way that can regulate serviceability and data coherence. The mechanism was incorporated to widely-used clustering algorithms, boosting their performance and usability in push systems. Furthermore, the incorporation retained all advantages of the clustering processes. Extensive simulations and comparison with alternatives verified the claims of the study. The presented mechanism can enable push services to attract more clients, while keeping the thematic data specialization in check, within acceptable bounds.

References

ACHARYA, S., ALONSO, R., FRANKLIN, M., AND ZDONIK, S. 1995. Broadcast disks. *ACM SIGMOD Record 24,* 2, 199–210.

ADOMAVICIUS, G. AND ZHANG, J. 2012. Impact of Data Characteristics on Recommender Systems Performance. *ACM Trans. Manage. Inf. Syst. 3,* 1, 3:1–3:17.

AL-SHAMRI, M. Y. H. AND BHARADWAJ, K. K. 2007. A Compact User Model for Hybrid Movie Recommender System. In *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*. 519–524.

ARTHUR, D. AND VASSILVITSKII, S. 2007. k-means++: the advantages of careful seeding. In *Proceedings of the 18th annual ACM-SIAM Symposium on Discrete Algorithms (SODA'07), New Orleans, Louisiana*. Society for Industrial and Applied Mathematics, 1027–1035.

BALLI, U., WU, H., RAVINDRAN, B., ANDERSON, J., AND JENSEN, E. 2007. Utility Accrual Real-Time Scheduling under Variable Cost Functions. *IEEE Transactions on Computers 56,* 3, 385–401.

BALTRUNAS, L., MAKCINSKAS, T., AND RICCI, F. 2010. Group Recommendations with Rank Aggregation and Collaborative Filtering. In *Proceedings of the Fourth ACM Conference on Recommender Systems*. RecSys '10. ACM, New York, NY, USA, 119–126.

BERKOVSKY, S. AND FREYNE, J. 2010. Group-based Recipe Recommendations: Analysis of Data Aggregation Strategies. In *Proceedings of the Fourth ACM Conference on Recommender Systems*. RecSys '10. ACM, New York, NY, USA, 111–118.

BEZDEK, J. C. AND HATHAWAY, R. J. 2002. Some Notes on Alternating Optimization. In *Advances in Soft Computing — AFSS 2002*, N. R. Pal and M. Sugeno, Eds. Lecture Notes in Computer Science, vol. 2275. Springer Berlin Heidelberg, Berlin and Heidelberg, 288–300.

BEZDEK, J. C. AND HATHAWAY, R. J. 2003. Convergence of alternating optimization. *Neural, Parallel Sci. Comput. 11,* 4, 351–368.

BIRANT, D. AND KUT, A. 2007. ST-DBSCAN: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering 60,* 1, 208–221.

CHIH-LIN HU AND MING-SYAN CHEN. 2009. Online Scheduling Sequential Objects with Periodicity for Dynamic Information Dissemination. *IEEE Transactions on Knowledge and Data Engineering 21,* 2, 273–286.

CLEGER-TAMAYO, S., FERNÁNDEZ-LUNA, J. M., AND HUETE, J. F. 2012. Top-N news recommendations in digital newspapers. *Knowledge-Based Systems 27*, 180–189.

CORLESS, R. M., GONNET, G. H., HARE, D. E. G., JEFFREY, D. J., AND KNUTH, D. E. 1996. On the LambertW function. *Advances in Computational Mathematics 5,* 1, 329–359.

CROSSEN, A., BUDZIK, J., AND HAMMOND, K. J. 2002. Flytrap: Intelligent Group Music Recommendation. In *Proceedings of the 7th International Conference on Intelligent User Interfaces*. IUI '02. ACM, New York, NY, USA, 184–185.

DAS, A. S., DATAR, M., GARG, A., AND RAJARAM, S. 2007. Google News Personalization: Scalable Online Collaborative Filtering. In *Proceedings of the 16th International Conference on World Wide Web*. WWW '07. ACM, New York, NY, USA, 271–280.

DASZYKOWSKI, M., WALCZAK, B., AND MASSART, D. 2001. Looking for natural patterns in data. *Chemometrics and Intelligent Laboratory Systems 56,* 2, 83–92.

ESTER, M., KRIEGEL, H.-P., SANDER, J., AND XU, X. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proc. of 2nd International Conference on Knowledge Discovery and*. 226–231.

ETZIONI, O., MULLER, J. P., BRADSHAW, J. M., AND KUSHMERICK, N. 1999. Learning to remove Internet advertisements. In *Proceedings of the third annual conference on Autonomous Agents - AGENTS '99*. ACM Press, 175–181.

FOULI, K., MAIER, M., AND MEDARD, M. 2011. Network coding in next-generation passive optical networks. *IEEE Communications Magazine 49,* 9, 38–46.

GARCIA, I., SEBASTIA, L., AND ONAINDIA, E. 2011. On the design of individual and group recommender systems for tourism. *Expert Systems with Applications 38,* 6, 7683–7692.

GROSSMAN, J., GROSSMAN, M., AND KATZ, R. 1980. *The First Systems of Weighted Differential and Integral Calculus*. Archimedes Foundation.

IKEDA, H., SUGAWA, J., ASHI, Y., AND SAKAMOTO, K. 2007. High-Definition IPTV Broadcasting Architecture Over Gigabit-Capable Passive Optical Network. In *IEEE GLOBECOM 2007-2007 IEEE Global Telecommunications Conference*. IEEE, 2242–2246.

ITU-T. 2010. 10-Gigabit-capable passive optical networks (XG-PON): Transmission convergence (TC) layer specification.

J. GECSEI. 1983. *The Architecture of Videotex Systems*. Prentice-Hall, Englewood Cliffs, NJ.

JAMESON, A. AND SMYTH, B. 2007. Recommendation to Groups. In *The Adaptive Web*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Lecture Notes in Computer Science, vol. 4321. Springer Berlin Heidelberg, Berlin, Heidelberg, 596–627.

JIANG, S. AND VAIDYA, N. H. 1999. Scheduling data broadcast to "impatient" users. In *Proceedings of the 1st ACM International Workshop on Data Engineering for Wireless and Mobile Access (MoBiDe'99), Seattle, Washington, United States*. ACM, 52–59.

JIANLIANG XU, XUEYAN TANG, AND WANG-CHIEN LEE. 2006. Time-critical on-demand data broadcast: algorithms, analysis, and performance evaluation. *IEEE Transactions on Parallel and Distributed Systems 17,* 1, 3–14.

KAKALI, V., SARIGIANNIDIS, P., PAPADIMITRIOU, G., AND POMPORTSIS, A. 2011. A Novel Adaptive Framework for Wireless Push Systems Based on Distributed Learning Automata. *Wireless Personal Communications 57,* 4, 591–606.

KANG, S. H., CHOI, S., CHOI, S. J., LEE, G., LEW, J., AND LEE, J. 2007. Scheduling Data Broadcast Based on Multi-Frequency in Mobile Interactive Broadcasting. *IEEE Transactions on Broadcasting 53,* 1, 405–411.

KIM, S. AND KANG, S. H. 2010. Scheduling Data Broadcast: An Efficient Cut-Off Point Between Periodic and On-Demand Data. *IEEE Communications Letters 14,* 12, 1176–1178.

KRIEGEL, H.-P., KRÖGER, P., AND ZIMEK, A. 2009. Clustering high-dimensional data. *ACM Transactions on Knowledge Discovery from Data 3,* 1, 1–58.

LEE, H. J. AND PARK, S. J. 2007. MONERS: A news recommender for the mobile web. *Expert Systems with Applications 32,* 1, 143–150.

LIASKOS, C. AND PAPADIMITRIOU, G. 2012. Entropy-based estimation of client preferences in wireless push systems. *IEEE Transactions on Communications 60,* 12, 3899–3908.

LIASKOS, C., PAPADIMITRIOU, G., NICOPOLITIDIS, P., AND POMPORTSIS, A. 2012. Parallel Data Broadcasting for Optimal Client Service Ratio. *IEEE Communications Letters 16,* 11, 1741–1743.

LIASKOS, C., PETRIDOU, S., AND PAPADIMITRIOU, G. 2011. Towards Realizable, Low-Cost Broadcast Systems for Dynamic Environments. *IEEE/ACM Transactions on Networking 19,* 2, 383–392.

LIASKOS, C., TSIOLIARIDOU, A., AND PAPADIMITRIOU, G. 2012. More for Less: Getting more clients by broadcasting less data. In *Proceedings of the 10th International Conference on Wired/Wireless Internet Communications (WWIC 2012), Santorini, Greece, June*. 64–75.

LIASKOS, C., TSIOLIARIDOU, A. N., AND PAPADIMITRIOU, G. 2013. On Data Compatibility and Broadcast Stream Formation. *IEEE Transactions on Computers 63,* 9, 2369–2375.

LIASKOS, C., XEROS, A., PAPADIMITRIOU, G., LESTAS, M., AND PITSILLIDES, A. 2012. Broadcast Scheduling with multiple concurrent costs. *IEEE Transactions on Broadcasting 58,* 2, 178–186.

MAN-SOO, H. 2012. Iterative dynamic bandwidth allocation for XGPON. In *Proceedings of the 14th International Conference on Advanced Communication Technology, Phoenix Park, PyeonhChang, Republic of Korea, February*. 1035–1040.

MASTHOFF, J. 2004. Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewers. *User Modeling and User-Adapted Interaction 14,* 1, 37–85.

MATHWORKS-INC. 2010. MATLAB version (R2010a).

MILLER, B. N., ALBERT, I., LAM, S. K., KONSTAN, J. A., AND RIEDL, J. 2003. MovieLens unplugged. In *the 8th international conference*, D. Leake, L. Johnson, and E. Andre, Eds. 263–266.

NICOPOLITIDIS, P., PAPADIMITRIOU, G., AND POMPORTSIS, A. 2002. Using learning automata for adaptive push-based data broadcasting in asymmetric wireless environments. *IEEE Transactions on Vehicular Technology 51,* 6, 1652–1660.

O'CONNOR, M., COSLEY, D., KONSTAN, J. A., AND RIEDL, J. 2001. PolyLens: A Recommender System for Groups of Users. In *Proceedings of the Seventh European Conference on Computer Supported Cooperative Work 16–20 September 2001, Bonn, Germany*. 199–218.

PAPADIMITRIOU, G., OBAIDAT, M., AND POMPORTSIS, A. 2002. On the use of learning automata in the control of broadcast networks: a methodology. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) 32,* 6, 781–790.

PENG, J., ZENG, D. D., AND HUANG, Z. 2008. Latent Subject-centered Modeling of Collaborative Tagging: An Application in Social Search. *ACM Trans. Manage. Inf. Syst. 2,* 3, 15:1–15:23.

PIETRONERO, L., TOSATTI, E., TOSATTI, V., AND VESPIGNANI, A. 2001. Explaining the uneven distribution of numbers in nature: The laws of Benford and Zipf. *Physica A: Statistical Mechanics and its Applications 293,* 1-2, 297–304.

PIZZUTILO, S., CAROLIS, B. D., COZZOLONGO, G., AND AMBRUOSO, F. 2005. Group Modeling in a Public Space: Methods, Techniques, Experiences. In *Proceedings of the 5th WSEAS International Conference on Applied Informatics and Communications*. AIC'05. World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 175–180.

PLUMBAUM, T., LOMMATZSCH, A., RUDNITZKI, S., LUCA, E. W., DWIGER, H., AND ALBAYRAK, S. 2010. Adaptive music news recommendations based on large semantic datasets. In *1st Workshop On Music Recommendation And Discovery (WOMRAD), ACM RecSys*.

POLIKAR, R. 2006. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine 6,* 3, 21–45.

RAISSI-DEHKORDI, M. AND BARAS, J. S. 2007. Broadcast Scheduling for Time-Constrained Information Delivery. In *Proceedings of the 2007 IEEE Global Telecommunications Conference (GLOBECOM'07), Washington, DC, USA, November*. 5298–5303.

RATTANAJITBANJONG, N. AND MANEEROJ, S. 2009. Multi criteria pseudo rating and multidimensional user profile for movie recommender system. In *2009 2nd IEEE International Conference on Computer Science and Information Technology*. 596–601.

RUI, H. AND WHINSTON, A. 2012. Designing a Social-broadcasting-based Business Intelligence System. *ACM Trans. Manage. Inf. Syst. 2,* 4, 22:1–22:19.

SCHONING, T. 1999. A probabilistic algorithm for k-sat and constraint satisfaction problems. In *40th Annual Symposium on Foundations of Computer Science*. 410–414.

SERPANOS, D. 2004. Scheduling objects in broadcast systems with energy-limited clients. *Computer Communications 27,* 10, 1036–1042.

UNIVERSITY OF CALIFORNIA IRVINE, SCHOOL OF INFORMATION AND COMPUTER SCIENCES. 1998. Machine learning repository: Internet advertisements data set, [online] https://archive.ics.uci.edu/ml/datasets/internet+advertisements.

UNIVERSITY OF EASTERN FINLAND, SCHOOL OF COMPUTING. 2012. Clustering datasets: S-sets, [Online] http://cs.joensuu.fi/sipu/datasets/.

UNNI, R. AND HARMON, R. 2007. Perceived effectiveness of push vs. pull mobile location-based advertising. *Journal of Interactive Advertising 7,* 2, 28–40.

VAIDYA, N. H. AND HAMEED, S. 1999. Scheduling data broadcast in asymmetric communication environments. *Wireless Networks 5,* 3, 171–182.

VALECHA, R., SHARMAN, R., RAO, H. R., AND UPADHYAYA, S. 2013. A Dispatch-Mediated Communication Model for Emergency Response Systems. *ACM Trans. Manage. Inf. Syst. 4,* 1, 2:1–2:25.

VARGA, A. AND FAKHAMZADEH, B. 1997. The k-split algorithm for the pdf approximation of multidimensional empirical distributions without storing observations. In *Proceedings of the 9th European Simulation Symposium (ESS'97), Passau, Germany*. 94–98.

YU, Z., ZHOU, X., HAO, Y., AND GU, J. 2006. TV Program Recommendation for Multiple Viewers Based on user Profile Merging. *User Modeling and User-Adapted Interaction 16,* 1, 63–82.

ZHAN, C., LEE, VICTOR C. S., WANG, J., AND XU, Y. 2011. Coding-Based Data Broadcast Scheduling in On-Demand Broadcast. *IEEE Transactions on Wireless Communications 10,* 11, 3774–3783.

ZHENG, B., WU, X., JIN, X., AND LEE, D. L. 2005. TOSA: a near-optimal scheduling algorithm for multichannel data broadcast. In *Proceedings of the 6th International Conf. on Mobile Data Management (MDM'05), Ayia Napa, Cyprus, May*. New York and USA, 29–37.