Marketing Science

My Mobile Music: An Adaptive Personalization System for Digital Audio Players

Tuck Siong Chung, Roland T. Rust, Michel Wedel,

My Mobile Music: An Adaptive Personalization System for Digital Audio Players

Tuck Siong Chung
Nanyang Business School, Nanyang Technological University, Singapore, 639798, atschung@ntc.edu.sg

Roland T. Rust, Michel Wedel
Robert H. Smith School of Business, University of Maryland, College Park, Maryland 20742
{rrust@rhsmith.umd.edu, mwedel@rhsmith.umd.edu}

New information technologies increasingly make it possible for service providers to adaptively personalize their service, fine-tuning the service over time for each individual customer, based on observation of that customer's behavior. We propose an "Adaptive Personalization System" and illustrate its implementation for digital audio players, a product category with rapidly expanding sales. The proposed system automatically downloads personalized playlists of MP3 songs into a consumer's mobile digital audio device and requires little proactive user effort (i.e., no explicit indication of preferences or ratings for songs). The system works in real time and is scalable to the massive data typically encountered in personalization applications. A simulation study shows the Adaptive Personalization System to outperform benchmark approaches. We implemented the Adaptive Personalization System on Palm PDAs and tested its performance with digital audio users. For actual users, the Adaptive Personalization System provides substantial improvements over benchmark approaches both in terms of the number of songs listened to and listening duration.

Key words: digital audio players; service marketing; personalization; customization; collaborative filtering; one-to-one marketing

History: This paper was received July 19, 2007, and was with the authors 1 month for 1 revision; processed by Steven Shugan. Published online in *Articles in Advance* May 15, 2008.

1. Introduction

Service is the fastest growing part of the world economy, and information service is the fastest growing part of service (Rust and Chung 2006). Information service offerings are expanding and competitive pressure is high. Consumer choice options continue to increase exponentially (in categories such as music, videos, vacations, news, and many more) as service providers seek to command higher margins by continuously differentiating, innovating, and expanding service offerings. Customizing services is seen increasingly as a key competitive strategy. Customization of service better meets individual customer needs and makes competitive retaliation more difficult (Sun 2006, Syam and Kumar 2006). As information technology advances, marketers are better able to interact with customers, store customer data, and process that data to personalize service. In line with this trend, recently, "Adaptive Personalization Systems" have begun to emerge that take full advantage of customer information to provide more personalized service (Ansari and Mela 2003, Rust and Chung 2006). These Adaptive Personalization Systems require minimal proactive user input, are based on observed purchase or usage data, and learn consumer tastes

adaptively over time. In this paper we propose such an Adaptive Personalization System and illustrate its application to music downloads for digital audio players.

Digital audio players are a fast growing information service. Mobile digital audio devices have become increasingly popular: more than one in ten American adults now owns an iPod or MP3 player.[1] However, the even faster growth of digital music offerings poses a daily challenge to consumers who want to find songs that fit their tastes. Music playlists have emerged as a tool to help consumers compile sets of songs they like. The Sprint Music Store was one of the first to offer over the air downloads in the United States. It is currently one of the market leaders with an estimated yearly volume of close to five million downloads. Their song database consists of over 1.5 million songs and the Sprint Music Manage Software© allows for browsing and purchasing songs over mobile broadband networks, (manual) downloading of songs on a personal computer, management of these songs, and transferal of the playlists from the computer to a cell phone.

---

[1] See Rainie and Madden (2005).

Several websites offer playlists that can be edited (e.g., Riffin, Smartplaylists) or that are customized based on ratings and reviews of songs (e.g., Music-mobs). While the playlists generated on these sites have to be listened to online, sites such as CDNow, MediaUnbound, and MoodLogic allow users to download into their mobile audio devices songs that are recommended based on attribute distance and user feedback. These sites take as inputs general questions on user music tastes, ratings of samples of music tracks, and open-ended questions. Other systems like Pandora and last.fm (recently bought by CBS, and having over 15 million customers) broadcast music to their Internet radio station listeners and customize their broadcasts based on user feedback and the similarity of song attributes. The Music Genome Project classifies songs on more than 400 "genetic" attributes (Castelluccio 2006). This database is used by Pandora to create personalized Internet radio stations for individual users based on the distance of songs in the attribute space, alternatively played through home music devices through Squeezebox© (Slim Devices) hardware. Rhapsody also uses the Genome Project as a basis for making music recommendations. It charges a monthly subscription fee, and users can listen to whatever they want online but have to pay for downloads and CDs. Lala.com has implemented a membership service that operates like a social network site and facilitates trading of CDs (currently, it allows users to also buy CDs). Consumers list the CDs they own on the website as well as CDs they would like to have, and the site makes trades, for a small fee per trade. This is possible because copying music is not legal, but trading it is.[2] Out of over 125 patents on recommendation/collaborative filtering systems filed between 1995 and 2005, 69% applied to music.[3]

The objective of the present study is to develop an Adaptive Personalization System for information services. The system is applied to mobile music and automatically downloads personalized playlists of MP3 songs into mobile digital audio devices. However, it can be extended to other information services as well. The system[4] works in real time, is scalable, and requires minimal proactive user effort. The personalization system is in spirit a music recommendation system, but it goes substantially beyond current music recommendation systems.

Recommendation systems generally fall into two categories: content filtering and collaborative filtering systems. Content filtering systems are agents that make recommendations based on the target individual's past preferences for product attributes and the similarity between products. Collaborative filtering, on the other hand, predicts an individual's preferences using other individuals' preferences. Memory-based systems use similarity measures, while model-based systems use mathematical and statistical techniques to predict preferences for recommended items. In the marketing literature, several model-based recommendation systems have been proposed (Ansari et al. 2000, Ying et al. 2006, Bodapati 2007). Research has shown that recommendation agents may influence user opinions (Cosley et al. 2003, Häubl and Murray 2003), reduce prices paid (Diehl et al. 2003), and improve the efficiency and quality of choices (Ariely et al. 2004, Häubl and Trifts 2000). Their effectiveness, however, depends on the extent to which they learn the user's decision rules (Aksoy et al. 2006). In addition, content and collaborative filtering-based recommendation agents have been shown to learn at different rates (Ariely et al. 2004). Furthermore, it has been shown that recommendations for unfamiliar items may receive less positive user evaluation, depending on the recommendation context (Cooke et al. 2002; this study also focused on music recommendations).

To enable personalization, input on individual music preferences is required. Although popular in practice, there are several problems associated with explicitly asking individuals for their music preferences. Consumers may be unwilling or unable to actively provide these evaluations and as a consequence most of the information in the systems in question is missing (Ying et al. 2006). In addition, when asked to provide an evaluation, individuals may respond idiosyncratically to the rating scale provided to them (Rossi et al. 2001). Furthermore, recommendation systems based on user input are sensitive to shilling attacks, in which crafted ratings are submitted to the system that cause it to make commercially desired recommendations (Lam and Riedl 2004). Finally, scalability presents an important challenge for all recommendation systems regardless of whether they use ratings input or not (Montgomery et al. 2004, Bodapati 2007). The databases used to generate recommendations are massive in terms of the number of individuals, choice alternatives, and their attributes. Massive data sets are an issue when the algorithm is computationally intensive, as with the Markov chain Monte Carlo (MCMC) procedures that have been used for much of the academic work in marketing on model-based recommendation systems (Ansari et al. 2000, Ying et al. 2006).

We propose an Adaptive Personalization System that alleviates many of the challenges summarized above. First, it learns music preferences based on how long songs have been listened to and thus does not depend on user ratings of songs. Second, we

---

[2] See Kuang (2007).

[3] Google Patent Search (May 2007).

[4] We have obtained a provisional patent protection for this system.

use an adaptive algorithm that updates the parameter estimates sequentially as blocks of listening data from new playlists come in. As more and more data become available, the system's adaptation to individual preferences improves. The algorithm is efficient and is scalable to massive data, because data are processed individual-by-individual and block-by-block. In addition, it incorporates a variable selection step that removes irrelevant song characteristics at the individual level, thus further reducing the system's computational burden. Finally, the proposed personalization system generates customized playlists using sequential design methods that trade off estimation precision and utility maximization over a finite time horizon. The personalization system downloads the playlists to a mobile audio device. Because it removes the need for individual inputs and adapts to individual changes in preferences, it enables music downloading to be fully automated.

The next section provides a description of the engine that is used to estimate individual preferences and customize playlists. Section 3 describes a small simulation study conducted as a preliminary test of our model. In §4 we provide a proof of concept of the system through a pilot implementation on PDAs with real users in their natural listening environment and a comparison with an alternative procedure. The paper ends with discussions, extensions, and conclusions.

2. The Adaptive Personalization System

This section describes the model used to estimate individual preferences and the personalization of individualized music playlists on the basis of these preferences. The flow diagram for the system is provided in Figure 1. The system includes:

(1) A Personalization Agent for Tastes—an individual-level hazard model used to estimate an individual's music tastes from the time he or she spent listening to a song;

(2) An Adaptive Learning Algorithm—a Bayesian variable selection procedure to select relevant song characteristics for each individual, and a particle filter for updating the models in real time as new data come in, making the system scalable to the massive data accumulated on song listening behavior;

(3) Collaborative Customization—a model averaging procedure in which models of different individuals are averaged weighted with their relative likelihoods; and

(4) Dynamic Customization—an optimal sequential experimental design procedure to generate customized utility maximizing playlists across a finite time horizon based on the model-averaged estimates.

The following subsections subsequently describe these four components of the recommendation system.

Figure 1   Flow Diagram of the Adaptive Personalization System
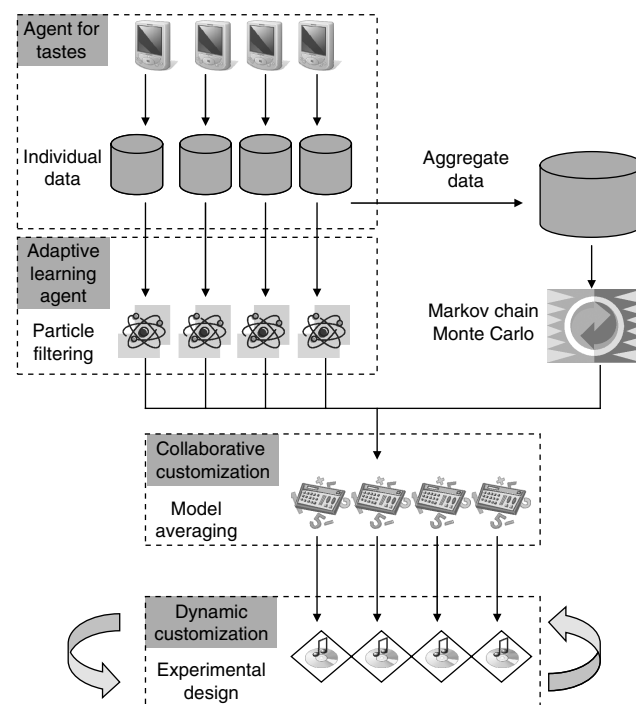
2.1. Personalization Agent: Lognormal Hazard Model of Listening Duration

The personalization agent calibrates its preference estimates separately for each individual, based on the recorded time an individual spends listening to previously generated song lists. Individual-level estimation enables real-time processing (and individual-level variable selection and collaborative customization, as will become apparent in the later subsections). Note that a hierarchical Bayes approach (Rossi and Allenby 2003) that has the attractive property of pooling information across individuals in estimation cannot be implemented, because its estimation cannot be parallelized across mobile devices, as we intend to do here. Thus, we estimate models at the individual level and pool the models in the prediction stage (through model averaging) rather than the estimation stage. We assume that the duration that an individual listens to a song is proportional to the utility he or she derives from that song. Utilities are inferred from the time an individual spends listening to a particular song,[5] using a hazard model. We utilize the lognormal distribution function for listening duration because it is flexible and its hazard is unimodal, the shape we expect for this type of data. Other hazard functions (Saha and Hilton 1997) represent these shapes as well, but the lognormal distribution has the additional, and

---

[5] This is less intrusive and requires less effort than the alternative method (e.g., Randall et al. 2007) of soliciting stated preferences from customers.

for our study crucial, characteristic that the mean is available in closed form, which facilitates computation in real time. The listening duration of individual i for song j, $y_{i,j}$, has a lognormal distribution with a mean μ_i and variance σ_i^2:

$$f(y_{i,j} \mid \mu_i, \sigma_i) = (2\pi)^{-1/2}(y_{i,j}\sigma_i)^{-1}$$
$$\cdot \exp\{-(\log(y_{i,j}) - \mu_i)^2/2\sigma_i^2\}. \quad (1)$$

The data are censored because we only observe the listening duration up to the length of the song in question, as reflected in the survival function given by

$$S(y_{i,j} \mid \mu_i, \sigma_i) = 1 - F[\{\log(y_{i,j}) - \mu_i\}/\sigma_i], \quad (2)$$

where $F(\cdot)$ is the normal cumulative distribution function (c.d.f.). The likelihood function takes the standard form

$$L(\mu_i, \sigma_i \mid \mathbf{y}_i) = \prod_{j=1}^{n_i} f(y_{i,j} \mid \mu_i, \sigma_i)^{\delta_j} S(y_{i,j} \mid \mu_i, \sigma_i)^{(1-\delta_j)}, \quad (3)$$

where n_i represents the number of observations for individual i, and δ_j is the censoring indicator for song j. Covariates are introduced through $\mu_i = \mathbf{x}'\boldsymbol{\beta}_i$. The matrix $\mathbf{x}'$ represents song attributes and genre dummies, and $\boldsymbol{\beta}_i$ contains the coefficients for individual i. The hazard function equals

$$f(y_{i,j} \mid \mu_i, \sigma_i)/S(y_{i,j} \mid \mu_i, \sigma_i),$$

the ratio of (1) and (2). The variance σ_i determines the shape of the lognormal hazard function. The hazard function is downward sloping for larger values of σ_i, so that it describes listening behavior in which an individual has decreasing tendency to stop listening to a song, reflecting increasing appreciation. For lower values of σ_i, the hazard rate increases to a maximum and then tapers off, capturing listening behavior of an individual needing certain duration of listening before he or she can decide that the song is unappealing.

Next to these individual-level models, we also use an aggregate model, calibrated on data obtained by stacking all individuals' listening durations to all songs. This model is used to capture the impact of a song's unique characteristics not otherwise reflected in the song characteristics in $\mathbf{x}'$. In addition, it facilitates the generation of playlists that contain novel attractive songs. The likelihood function is

$$L(\mu, \sigma \mid \mathbf{y}) = \prod_{i=1}^{N}\prod_{j=1}^{n_i} f(y_{i,j} \mid \mu, \sigma)^{\delta_{i,j}} S(y_{i,j} \mid \mu, \sigma)^{(1-\delta_{i,j})}, \quad (4)$$

where the distribution and survivor functions are similar to (1) and (2). We specify $\mu = \mathbf{z}'\boldsymbol{\alpha}$, where $\mathbf{z}'$ is a vector consisting of a dummy variable for every song,

and $\boldsymbol{\alpha}$ is the corresponding vector of coefficients, and the vector $\mathbf{y}_i = (y_{i,j})$.

For the first block of listening data from every individual, we use the Metropolis-within-Gibbs algorithm, which includes a variable selection step, to sample from the posterior distribution. This first block of data is obtained from the first playlists that the individuals listen to. Without the variable selection step, MCMC sampling involves successively drawing from the full conditional posteriors $p(\boldsymbol{\beta}_i \mid \sigma_i^2, \boldsymbol{\Omega}_{\boldsymbol{\beta}_i}, \mathbf{y}_i)$, $p(\boldsymbol{\Omega}_{\boldsymbol{\beta}_i} \mid \boldsymbol{\beta}_i)$, and $p(\sigma_i^2 \mid \boldsymbol{\beta}_i, \mathbf{y}_i)$, where $\boldsymbol{\Omega}_{\boldsymbol{\beta}_i}$ represents the variance-covariance matrix of individual i's coefficients. The variable selection step is detailed in §2.2.

For the aggregate model we can only estimate a parameter for a particular song when at least one individual has listened to that song. Therefore, we need to estimate new song-specific constants as new data comes in, because some individuals have now listened to new songs. Consequently, we run a new MCMC chain to estimate these new song-specific constants. Only the most recent listening duration of an individual for a particular song is used, which keeps the size of the data set manageable and at the same time accommodates changes in music tastes, because the constants are estimated using only the most recent data.[6]

2.2. Adaptive Learning: Bayesian Variable Selection of Song Characteristics

Typical applications of our personalization system have a large number of explanatory variables. This is especially the case when the system includes song attributes and genres. For any individual, many of the explanatory variables may be redundant. Variable selection is applied to the individual models to resolve this problem. It is not used for the aggregate model, because in that model we desire estimates of all song-specific constants. Conceptually, using a variable selection procedure alleviates the assumption that music tastes are formed through compensatory processes. Variable selection implies a partially noncompensatory approach that assumes that individuals consider only a subset of attributes, while preference formation is compensatory among the attributes that are considered (Gilbride and Allenby 2006). We prefer to use variable selection over alternative parsimonious representations such as factor representations (Haaijer et al. 1998, Ansari and Jedidi 2000),

---

[6] The MCMC algorithm for the aggregate model may, depending on the size of the customer database, present problems for the scalability of the model. Several solutions are possible. We recommend ignoring the censoring, and using conjugate priors (Rossi et al. 2005, pp. 22–26), so that the posterior distributions are available in closed form. Samples from the posteriors can then be obtained directly without iteration, even for massive data sets, using the mean log-listening durations as sufficient statistics.

for its interpretation as a noncompensatory preference formation model (Gilbride and Allenby 2006) and for its greater computational attractiveness that arises because eliminated attributes do not need to be kept in memory (George and Foster 2000).

The goal of variable selection is to eliminate the pth attribute for song j, $x_{j,p}$ if $\beta_{i,p}$ is close to zero (Chipman et al. 2001). This involves selecting a sub-model, based on the lognormal likelihood for $\mathbf{y}_i$ provided in (3): $L(\mathbf{x}_{i,\psi_i}\boldsymbol{\beta}_{i,\psi_i}, \sigma_i^2 \mathbf{I} \mid \mathbf{y}_i)$, where $\boldsymbol{\psi}_i = (\psi_{i,1}, \ldots, \psi_{i,p})$ are indicator variables that specify if $\beta_{i,p}$ is set to zero ($\psi_{i,p}=0$), or not ($\psi_{i,p}=1$), and $\boldsymbol{\beta}_{i,\psi_i}$ is the reduced vector of regression coefficients, and $\mathbf{x}_{i,\psi_i}$ is the reduced reduced x-matrix corresponding to a particular $\boldsymbol{\psi}_i$.

The vector $\boldsymbol{\psi}_i$ is updated via a Metropolis step during estimation. The full MCMC chain involves successively drawing from the following posterior distributions:

$$
\begin{aligned}
&p(\boldsymbol{\psi}_i^{\text{new}} \mid \boldsymbol{\beta}_{i,\psi_i}, \sigma_i^2, \boldsymbol{\psi}_i^{\text{old}}, \mathbf{y}_i), \\
&p(\boldsymbol{\beta}_{i,\psi_i} \mid \sigma_i^2, \boldsymbol{\Omega}_{\boldsymbol{\beta}_{i,\psi_i}}, \boldsymbol{\psi}_i, \mathbf{y}_i), \\
&p(\boldsymbol{\Omega}_{\boldsymbol{\beta}_{i,\psi_i}} \mid \boldsymbol{\beta}_{i,\psi_i}, \boldsymbol{\psi}_i), \\
&p(\sigma_i^2 \mid \boldsymbol{\beta}_{i,\psi_i}, \boldsymbol{\psi}_i, \mathbf{y}_i).
\end{aligned}
\tag{5}
$$

The variable selection indicator vector $\boldsymbol{\psi}_i$ is updated by randomly changing one of the P indices in $\boldsymbol{\psi}_i^{\text{old}}$ (e.g., $\psi_{i,p}^{\text{old}}$). If $\psi_{i,p}^{\text{old}}=1$, we set $\psi_{i,p}^{\text{new}}=0$. If $\psi_{i,p}^{\text{old}}=0$, we set $\psi_{i,p}^{\text{new}}=1$ and assign $\beta_{i,p}$ a new nonzero value taken from $\beta_{i,\psi^{\text{old}}}$ (Sha et al. 2006). The proposed $\boldsymbol{\psi}_i^{\text{new}}$ is accepted with probability

$$
\min\left\{1, \frac{L(\boldsymbol{\beta}_{i,\psi_i^{\text{new}}}, \sigma_i^2, \boldsymbol{\psi}_i^{\text{new}} \mid \mathbf{y}_i)}{L(\boldsymbol{\beta}_{i,\psi_i^{\text{old}}}, \sigma_i^2, \boldsymbol{\psi}_i^{\text{old}} \mid \mathbf{y}_i)}\right\}.
\tag{6}
$$

The form of the posterior distributions for $\boldsymbol{\beta}_{i,\psi_i}$, $\boldsymbol{\Omega}_{\boldsymbol{\beta}_{i,\psi_i}}$, and σ_i^2 is standard. We use the following priors for $\boldsymbol{\beta}_{i,\psi_i}$, $\boldsymbol{\Omega}_{\boldsymbol{\beta}_{i,\psi_i}}$, and σ_i^2:

$$
\begin{aligned}
&p(\boldsymbol{\beta}_{i,\psi_i} \mid \sigma_i^2, \boldsymbol{\Omega}_{\boldsymbol{\beta}_{i,\psi_i}}, \boldsymbol{\psi}_i, \mathbf{y}_i) = N(0, \boldsymbol{\Omega}_{\boldsymbol{\beta}_{i,\psi_i}}), \\
&p(\boldsymbol{\Omega}_{\boldsymbol{\beta}_{i,\psi_i}} \mid \boldsymbol{\beta}_{i,\psi_i}, \boldsymbol{\psi}_i) = IG(v/2, v\lambda^x/2), \\
&p(\sigma_i^2 \mid \boldsymbol{\beta}_{i,\psi_i}, \boldsymbol{\psi}_i, \mathbf{y}_i) = IG(v/2, v\lambda^y/2).
\end{aligned}
\tag{7}
$$

We choose a small value $v=5$ for a diffuse prior on $\boldsymbol{\Omega}_{\boldsymbol{\beta}_{i,\psi_i}}$ and σ_i^2, and equate λ^x to the sample variance of $\mathbf{x}_i$, and λ^y to the sample variance of $\mathbf{y}_i$. After a burn-in period, the draws for the values of $\boldsymbol{\beta}_{i,\psi_i}$, σ_i, and $\boldsymbol{\psi}_i$ are saved as initial particles used for the Sequential Monte Carlo procedure described in the next section.

2.3. Adaptive Learning: Particle Filter for Updating of Parameter Estimates

Standard MCMC involves a large number of iterations, each of which requires a complete scan of the data set. In addition, if new data come in, a full scan of all available data is again needed. This makes Bayesian analysis of massive and sequentially accumulated data sets, as in our personalization system, using standard MCMC infeasible. We therefore use sequential Monte Carlo methods, or particle filters, to estimate the model parameters for the individual models (Ridgeway and Madigan 2003). Not only does this procedure allow us to analyze large data sets, it also enables us to efficiently update parameters as new blocks of data come in, which is crucial for Adaptive Personalization Systems. The basic idea is as follows. The observed data for individual i are obtained sequentially in batches, ${}^1\mathbf{y}_i, {}^2\mathbf{y}_i, \ldots, {}^K\mathbf{y}_i$; $k=1, \ldots, K$ denotes the batch number and ${}^k\mathbf{y}_i = ({}^k y_{i,1}, \ldots, {}^k y_{i,j})$. The particle filter starts from $m=1, \ldots, M$ initial values of the parameters $\boldsymbol{\beta}_{i,\psi_i}$, σ_i, and $\boldsymbol{\psi}_i$ (collected in ${}^1\boldsymbol{\Theta}_i^m$) that are draws from the posterior distribution of the parameters given the first wave of data ${}^1\mathbf{y}_i$. These are used to generate samples (particles) ${}^2\boldsymbol{\Theta}_i^m$ from the posterior distribution of the parameters that includes the information from the second wave of data ${}^2\mathbf{y}_i$. But instead of generating new samples from that posterior as would be done in standard MCMC methods, the previous particles ${}^1\boldsymbol{\Theta}_i^m$ are reweighted. The weights ${}^2w_i^m$ are derived from the likelihood of the model, where the vector of weights ${}^2\mathbf{w}_i$ is calculated based only on ${}^2\mathbf{y}_i$, and ${}^3\mathbf{w}_i$ is calculated based on ${}^3\mathbf{y}_i$ and so on. Resampling of the particles is needed each time a large number of them has a weight close to zero.

Thus, once we have estimated individual parameters using MCMC on the first block of data, ${}^1\mathbf{y}$, we retain the draws after a burn-in period as M particles. Each particle is a vector of draws from the posterior distributions of $\boldsymbol{\beta}_{i,\psi_i}$, σ_i, and $\boldsymbol{\psi}_i$, denoted by $\boldsymbol{\Theta}_i^m$. The particles are used to approximate the full conditional distribution of the parameters $P(\boldsymbol{\Theta}_i \mid {}^1\mathbf{y}_i)$. Note that variable selection results in a substantial dimensionality reduction of the number of particles for each individual, and note that the coefficients for the variables selected vary across the M particles. We focus on the second wave of data first. When that second block of data arrives after individuals have listened to the second playlist, the estimates of the individuals' parameters need to be updated. Now an approximation to the full conditional posterior $P(\boldsymbol{\Theta}_i \mid {}^1\mathbf{y}_i, {}^2\mathbf{y}_i)$ is given by

$$
P(\boldsymbol{\Theta}_i \mid {}^1\mathbf{y}_i, {}^2\mathbf{y}_i) \approx \sum_{m=1}^{M} w_{i,m}\delta(\boldsymbol{\Theta} - \boldsymbol{\Theta}_i^m) \Big/ \sum_{m=1}^{M} w_{i,m}.
\tag{8}
$$

Here, $\delta(\cdot)$ is the *dirac* delta measure and the $w_{i,m}$ are importance sampling weights (Ridgeway and Madigan 2003) that are proportional to the likelihood:

$$
w_{i,m} \propto \prod_{j=1}^{J} L({}^2 y_{i,j} \mid \boldsymbol{\Theta}_i^m).
\tag{9}
$$

Thus, as the second wave of data comes in, the particles obtained after the first wave are updated with importance weights that are proportional to the likelihood of the second-wave data. A potential problem with this method of updating is that the posterior variance of the parameters conditioned on $^1\mathbf{y}_i$ and $^2\mathbf{y}_i$ will be smaller than that conditioned on $^1\mathbf{y}_i$ alone. The more concentrated the posterior distribution is after the second block of data is accounted for, the larger the proportion of particles is that has near-zero weight. This reduces the efficiency of the importance weighting, which can be mitigated by incorporating a rejuvenation step. The rejuvenation step is invoked when the effective sample size (ESS) falls below 10% (Ridgeway and Madigan 2003). Kong et al. (1994) show that

$$\mathrm{ESS} = \left(\sum_{m=1}^{M} w_{i,m} \right)^2 \Big/ \sum_{m=1}^{M} (w_{i,m})^2. \qquad (10)$$

We carry out the rejuvenation step using the systematic resampling procedure proposed by Arulampalam et al. (2002). This procedure involves uniformly sampling the particles M times with replacement, using probabilities derived from the particle weights: particles with larger weights will be selected more often. The new particles are each given a weight of $1/M$. As a new wave k of listening data comes in, Equations (8) and (9) are applied again so that the particles $^k\Theta_i^m$, and the weights $^k w_{i,m}$ are available to approximate the posterior distribution $P(\Theta_i \mid {}^1\mathbf{y}_i, \ldots, {}^k\mathbf{y}_i)$. The expectation of a statistic of the parameters $h(\Theta_i)$, in particular, the mean expected listening duration for individual i that we use in the playlist design, can be computed from the particles after each wave as

$$\hat{E}(h(\Theta_i) \mid {}^1\mathbf{y}_i, \ldots, {}^k\mathbf{y}_i) \approx \sum_{m=1}^{M} h({}^k\Theta_i^m)/M. \qquad (11)$$

In the empirical application, we will use $M = 500$ particles. The convergence of the particles is assured under weak assumptions as $M \to \infty$, at a rate of $1/M$ (Crisan and Doucet 2002). In practical applications with a finite number of particles, procedures for assessing convergence of the particle filter are not available. However, the simulation results that we present in §3 indicate that the number of particles used is sufficient for good parameter recovery.

2.4. Collaborative Customization: A Model-Averaging Approach

Combining predictions of listening durations across models for different individuals will improve our predictions of a target individual's listening duration, in particular for songs that the individual has not listened to. We call this collaborative customization

because it is similar in spirit to collaborative filtering. It involves combining individuals' models, as well as the aggregate model, based on how well they predict the target individuals' listening data. To achieve this, we use model averaging (Hoeting et al. 1999). The basic idea is that by averaging over many different models, model uncertainty is incorporated into the predictions, and this always provides better predictions than using any of the models in isolation (Madigan and Raftery 1994). Bayesian model averaging involves the computation of the predictive distribution of $\hat{y}$, the predicted future value of the dependent variable, across I models as

$$p(\hat{y} \mid \mathrm{Data}) = \sum_{i=1}^{I} p(\hat{y} \mid \mathrm{Model}_i, \mathrm{Data}) \times p(\mathrm{Model}_i \mid \mathrm{Data}).$$

In other words, the predictive distribution of $\hat{y}$ is calculated as a weighted average of the distributions of $\hat{y}_j$ for each of the different models considered, weighted by these models' posterior probabilities. If we assume that all the models have prior probability, $p(\mathrm{Model}_i) = 1/I$, then we have

$$\frac{p(\mathrm{Model}_i \mid \mathrm{Data}) = p(\mathrm{Data} \mid \mathrm{Model}_i)}{\sum_{i=1}^{I} p(\mathrm{Data} \mid \mathrm{Model}_i)}.$$

Therefore, the posterior probability of model i is given by the ratio of its likelihood over the sum of likelihoods for all models.

Thus, specifically we compute a weighted average forecast of models for a particular individual i, where the weights used are posterior odds assuming a uniform prior $(1/I)$ for the models of all individuals. The weight applied to individual l's prediction of the expected listening duration for individual i at wave k is (Hoeting et al. 1999)

$$^k\phi_{i,l} = \frac{L(\Theta_l \mid {}^k\mathbf{y}_i)}{L(\Theta_i \mid {}^k\mathbf{y}_i)}. \qquad (12)$$

These weights are calculated for all individuals $l = 1, \ldots, I$, and the aggregate model, $I+1$. These weights can be computed very quickly because they involve closed-form expressions. In addition, strategies exist for reducing the number of models considered by truncating the posterior model odds below and above (at, e.g., 0.05 and 1). In most cases the number of different models considered is between 10 and 100, which greatly reduces the computational burden (Madigan and Raftery 1994). The model-averaged prediction of the log-listening duration of individual i for song j in wave k is then

$$^k\hat{y}_{i,j} = \sum_{l=1}^{I+1} {}^k\xi_{l,j} \times {}^k\phi_{i,l} \times {}^k\hat{y}_{i,j}^l \Big/ \sum_{l=1}^{I+1} {}^k\xi_{l,j} \times {}^k\phi_{i,l}. \qquad (13)$$

Here, ${}^k\hat{y}^l_{i,j}$ is the prediction of individual i's log-listening duration in wave k using a weighted average of the M particles for the parameters of individual l:

$$ {}^k\hat{y}^l_{i,j} = \left(\sum_{m=1}^{M} {}^k w_{l,m} \big(\mathbf{x}'^k_j \boldsymbol{\beta}^m_{l,\boldsymbol{\psi}_i} + ({}^k\sigma^m_l)^2/2 \big) \middle/ \sum_{m=1}^{M} {}^k w_{l,m} \right). \quad (14) $$

In Equation (13), ${}^k\xi_{i,j} = 1$ if individual i has ever listened to song j in wave k or before; ${}^k\xi_{i,j} = 0$ otherwise. In addition, ${}^k\xi_{I+1,j} = 1$ if at least one individual has ever listened to song j; ${}^k\xi_{I+1,j} = 0$ otherwise. These indicator variables ensure that predictions for a song are only based on models for other individuals that have ever listened to the song in question.

2.5. Generating Playlists: Bayesian Sequential Experimental Design

We use an experimental design method to determine the composition of the playlists. This approach enables us to choose the composition of songs in our playlists to maximize consumer listening, improve future parameter estimates, and facilitate the occasional generation of surprisingly interesting songs. A Bayesian approach is used to account for uncertainty in the parameter estimates by integrating over their distribution in computing the design criterion. Playlists are personalized and adapted each time new data on individual's preferences are collected, using a sequential design generating procedure (Pronzato and Thierry 2003).

The songs in the initial playlist are selected randomly from the song database. The reason for doing this is that it enables us to better estimate the initial subject-specific parameters that are needed for the construction of the subsequent playlists. A random playlist has more variability in the song attributes than songs that are selected by the respondent and will not suffer from endogeneity; that is, the attributes are uncorrelated with the individuals' listening duration error, which may not be the case if songs chosen by the user are used. An alternative start-up approach, starting from user ratings or a user-supplied song list, might instead be employed.

Maximizing expected listening duration has the disadvantage that the design quickly converges to a similar playlist on all future occasions. That makes some parameters inestimable, because subsets of songs that enable the identification of parameters (i.e., genres) no longer appear in the playlist. Therefore, we dynamically optimize both expected listening duration and parameter efficiency over a finite time horizon. Including efficiency in the optimization criterion spreads the design of the playlist across the song attributes, enables estimation of the parameters

(Verdinelli and Kadane 1992), and promotes serendipity that enables individuals to find surprising and interesting songs in the playlists. Each time we obtain new data on the individuals' listening behavior we customize new playlists. Because the playlist involves a set of $q = 1, \ldots, Q$ songs, it requires a batch sequential design procedure. We generate optimal playlists for individual i in wave k by maximizing the criterion:

$$ E\left[\sum_{q=1}^{Q} {}^k\hat{y}_{i,q} + [\log\{(N_d - k - 1) \times Q + 1\}]^2 \right. $$

$$ \left. \cdot \log \left| \mathbf{x}'_{d_{\boldsymbol{\psi}_i}} \mathbf{x}_{d_{\boldsymbol{\psi}_i}} + R_{\boldsymbol{\psi}_i} / (\sigma^k_i)^2 \right| \right]. \quad (15) $$

Here, ${}^k\hat{y}_{i,q}$ is the predicted log-listening duration for song q in wave k based on the model averaging (Equation (14)) and $\log|\mathbf{x}'_{d_{\boldsymbol{\psi}_i}} \mathbf{x}_{d_{\boldsymbol{\psi}_i}} + R_{\boldsymbol{\psi}_i}/(\sigma^k_i)^2|$ is the design efficiency (Verdinelli and Kadane 1992). In the design matrix $\mathbf{x}_{d_{\boldsymbol{\psi}_i}}$ and the prior precision matrix $R_{\boldsymbol{\psi}_i}/(\sigma^k_i)^2$, the variables that are taken out of the model during the variable selection process are eliminated, as indicated by the subscript $\boldsymbol{\psi}_i$. The expectation in Equation (15) is taken with respect to the uncertainty in the parameters, and is approximated by an average across the M particles obtained as output from the particle filter, as in Bayesian designs (Sandor and Wedel 2001).

The term $[\log\{(N_d - k - 1) \times Q + 1\}]^2$ is a weight that captures the sequential trade-off between precision and listening duration. This weight increases the "spread" of the design to continue to provide information on the coefficients, even as the playlists are successively zoomed in on the individuals' preferences. It ensures that a certain importance is placed on getting more accurate estimates of the parameters from the first playlists. As the estimates improve, subsequent playlists are generated by placing greater weight on the predicted listening duration. $k - 1$[7] is the number of playlists that have been generated for the individual so far, and N_d is the number of designs that will be created before the weight drops to zero. In other words, the design is optimized across a finite time horizon of length N_d. If the weight is zero, the optimal playlist maximizes the expected listening duration, $E[{}^k\hat{y}_{i,q}]$. The value of N_d can be chosen by the user and enables the playlist generation to restart periodically, which may be desirable. It is advantageous for the value of N_d to be somewhat large so that the playlists do not quickly become very similar. However, too large a value causes a lower overall predicted listening duration of the songs in the initial design, which may render the playlists less attractive.

---

[7] $k - 1$ is used instead of k because the playlists are customized from wave two (i.e., $k = 2$) onwards.

Although a value of N_d may be subjectively chosen, from our simulation studies, $N_d = 10$ appears to be a reasonable choice.

The optimal playlist is constructed using a modified Fedorov algorithm (Cook and Nachtsheim 1980). This algorithm is fast and very well suited for optimal design problems (Kuhfeld et al. 1994); is easy to implement, robust, insensitive to the initial design; and performs at least as well as several alternative procedures (Ogungbenro et al. 2005). It operates as follows. A starting playlist is derived by selecting songs randomly from the song database (but alternatively songs already stored in the PDA by the user can be used to initialize the procedure). The algorithm then proceeds iteratively. At the first iteration, the algorithm exchanges the first song in the current playlist with the first song from the song database. If this yields an improvement of the criterion (15) for that individual, the new song is included. If not, the song in the current playlist is exchanged with the next song in the database. This process continues until all songs in the database have been tried. Then, the algorithm moves to the next song on the playlist and exchanges each possible song in the song database with it to see if that yields an improvement in the criterion. This iterative replacement method is carried out until there is no further improvement after a complete pass through all songs in the playlist. In the design of playlists, only their composition is optimized, not their size (which is fixed at Q), nor the order of the songs, which is random.

3. Test on Synthetic Data

As an initial test of our personalization system, we ran a simulation study to investigate the effectiveness of the variable selection and model-averaging procedures. Furthermore, we compared the performance of our system with those of three other approaches. Because in adaptive personalization we are interested in how well the system performs for every individual, we intend to investigate all individual-level results in detail and therefore focus our attention on five simulated individuals (we use a larger sample in the pilot implementation in the next section). The song attributes used in the simulated data comprise genres and musical attributes.

The genres are Alternative/Indie, Electronic/Dance, Jazz, Metal, Pop, Rap/Hip Hop, Reggae/Ska, Rock, Soul/R&B, or Vocal. The music attributes include the perceived loudness, tempo, instrumental/vocal, solo/orchestra, listening/dancing, and happy/angry mood. We obtained these attributes for the 400 actual songs that will also be used for the implementation described below. The attributes are obtained from http://www.musiclens.de.

We generated playlists of 50 songs. We assumed that each simulated individual listens to each song 10 times, so that each playlist generates 500 observations. For every individual, we generated listening durations for each of these 500 observations based on an assumed set of parameter values for each of the attributes and genres (not shown). We estimated the model using these data and customized subsequent playlists based on these estimates, following the procedures described above. For new playlists, listening durations were generated with the same set of parameter values for each individual. Censoring of the listening durations was kept between 25%–35%. For the initial block of data 2,500 MCMC iterations were used with a burn-in of 2,000 iterations. This provided us with 500 particles. The results of the estimation of the individual parameters are shown in Table 1, both with and without variable selection, after the first customized playlist. The table shows the hit rate (percentage of variables correctly selected), and the root mean-squared errors (RMSE) for nonzero parameters. The variable selection works very well, effectively forcing nonsignificant parameters to zero and keeping significant ones in, in over 90% of the cases. Without variable selection the parameter estimates are more biased, because coefficients are estimated for nonessential variables. Table 1 shows that for all individuals parameter recovery is quite good and substantially better than without variable selection.

The RMSE of the predicted listening duration for all 400 songs with and without model averaging are shown in Table 2, based on data from the first customized playlist. Table 2 shows that the improvement from model averaging is substantial. For all simulated

Table 1   Selection Hit Rate and RMSE of Parameter Recovery on Synthetic Data With and Without Variable Selection

	Individual 1	Individual 2	Individual 3	Individual 4	Individual 5
Variable selection					
Hit rate (%)	100	100	100	94	94
RMSE	0.033	0.039	0.124	0.095	0.057
No variable selection					
Hit rate (%)	17	22	39	39	39
RMSE	0.061	0.081	0.099	0.103	0.124

Table 2 **RMSE of Listening Duration Predictions on Synthetic Data With and Without Model Averaging**

	Individual 1	Individual 2	Individual 3	Individual 4	Individual 5
RMSE with model averaging	1.82	1.67	2.57	4.09	1.91
RMSE without model averaging	2.52	1.86	3.33	6.58	2.17

Table 3 **Comparison of System Performance on 10 Waves of Synthetic Data in Terms of Duration of Listening**

	Individual 1	Individual 2	Individual 3	Individual 4	Individual 5
Adaptive Personalization System (new) Total duration	416.29	318.22	400.09	465.89	357.98
Maximum duration (new) Total duration	458.33	318.22	400.09	484.31	361.87
Minimum dissimilarity Total duration	306.10	281.49	267.63	289.73	271.20
Random generation Total duration	229.09	153.25	173.62	214.88	187.22

individuals model averaging even outperforms models that are calibrated on data generated from that same model. These results make a strong case for collaborative customization as applied in the Adaptive Personalization System.

We compare the performance of the optimal design against three other designs: (1) *maximum predicted duration* (new in this study), (2) *minimum dissimilarity* (a benchmark similar to the Pandora.com approach), and (3) *random generation* (a naïve benchmark model). The first design, maximum predicted duration, generates a playlist that maximizes only the listening duration predicted by model averaging and does not make the trade-off with precision of the parameter estimates. This procedure will by definition provide longer predicted listening durations but serves as a useful benchmark to assess the performance of our procedure versus the maximum attainable. The second procedure, minimum dissimilarity, generates playlists by minimizing the average Gower distances[8] of the songs in the playlist from the songs in the last block that the individual in question finished listening to. This benchmark is similar in spirit to the item-to-item recommendation system used by Pandora.com. The third design, random generation, generates a playlist by choosing songs randomly.

Table 3 shows the performance of the optimal design after generating 10 playlists for each individual. In most cases, the predicted listening duration of

the optimal design is somewhat lower than that of the design that maximizes predicted duration, because the optimal design trades off duration for precision in the initial playlists. However, the predicted listening durations of these two procedures are fairly close in most cases, in part due to the fact that the proposed design is optimized sequentially over a sequence of 10 playlists. The minimum dissimilarity design gives a much lower predicted listening duration. We conclude that the procedure proposed in this study works as expected and performs well compared to benchmark methods with respect to listening duration.

4. Pilot Implementation

4.1. Design and Data Collection

We provide a pilot implementation of the Adaptive Personalization System. The experiment is designed to have individuals listen to the songs played from a playlist on a Palm TX PDA. Such a system did not previously exist and was developed from scratch. A program was written for the PDAs to enable playing the MP3 songs from a playlist and to enable collection and storage of listening data. The PDAs were programmed in the Handheld Basic Software developed by Peter Holmes Consulting. An R-program was developed to implement the Adaptive Personalization System and the benchmark algorithms on a desktop computer.[9] The PDA and screen shots of the PDAs program are shown in Figure 2.

---

[8] The Gower (1971) distance is one of the most popular measures of similarity (or distance) between a pair of individuals based on a set of variables of a mixed data type (i.e., continuous and categorical variables). It is defined as the average of the absolute distance between two individuals on each of the variables, divided by the range of the variables in the data.

[9] Relative to this system two improvements have been tested. First, downloading of recommendations and uploading of data can be done over the Internet if the PDA has an Internet connection. Second, the particle filter that estimates individual-level parameters can be run on the programmable Palm PDA.

The user interface (UI) for the system is simple. Figure 3 shows the UI-flowchart. When the user connects to the server, data are uploaded, and personalized playlists are generated and downloaded from the server to the PDA. The playlists are played in randomized order, and the user can decide to skip over a song at any time. In addition, users also have the options of pausing, resuming, and stopping the song. If a song is skipped or paused, the duration is recorded and the system flags it as being uncensored. If a song is listened to completely, the system flags the duration as censored (because duration is truncated by the length of the song). These are the data stored for every song in the PDA and uploaded to the server. Each playlist holds 50 different MP3 songs out of the universal set of 400 unique MP3 songs used for the experiment.

To collect the duration data while the individuals were in a natural situation, respondents took the PDAs with them and were at liberty to decide when and where to listen to the songs. There were two waves of listening for each individual, with each wave lasting five days. After the first-wave data were uploaded from the PDA and the playlists customized for the second wave. In each wave songs were played by the PDAs from the customized playlists; the playlists used for the first wave were generated randomly.

About one-third of the individuals were randomly chosen to be in the control group and were subjected to the same procedures as the experimental group, except that their playlists were customized using the minimum dissimilarity procedure. The minimum dissimilarity model entails a heuristic item-to-item personalization algorithm. This algorithm generates a playlist for the target individual by choosing the songs that are most similar to the songs the individual finished listening to during the first wave. Similarity between songs in the song-attribute space

was measured by Gower distances. The procedure minimized the sum of the Gower distances of the 50 songs in the playlist to the target songs in the first wave. Such a heuristic procedure is conceptually similar to the one used by Pandora.com, but note that even an item-to-item system has not been developed for downloading songs into mobile audio devices as yet. Playlists in the experimental group were customized using the proposed Adaptive Personalization System algorithm. The song attributes used were obtained from www.musiclens.de and are the same as the ones described above for the simulated data. The genres were Alternative/Indie, Electronic/Dance, Jazz, Metal, Pop, Rap/Hip Hop, Reggae/Ska, Rock, Soul/R&B, or Vocal; and music attributes were loudness, tempo, instrumental/vocal, solo/orchestra, listening/dancing, and happy/angry mood. We obtained these attributes for all 400 songs in our music database.

Participants in the experiment were undergraduate and graduate students from an Eastern U.S. University who volunteered to participate between April and May 2007. Respondents who finished the experiment were given a monetary reward in addition to the chance of winning a Palm TX PDA. The 86 individuals who participated in the experiment were solicited without screening based on demographic criteria. Of the 86 individuals 77% were between 18 to 21 years old, 20% were aged 22 to 29 years old, while the others were 30 years or older. In addition, 37% of the individuals were male, while 63% of the individuals were female. These demographics are fairly representative of the target population of mobile music personalization systems. Respondents were informed that their listening behaviors would be recorded. The data captured were records of individuals' listening durations for the songs played from the PDAs: 16,835 data points were collected.

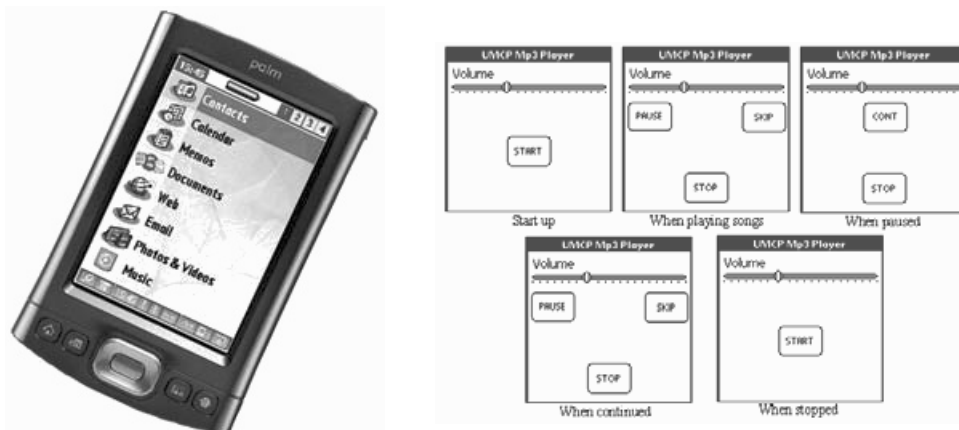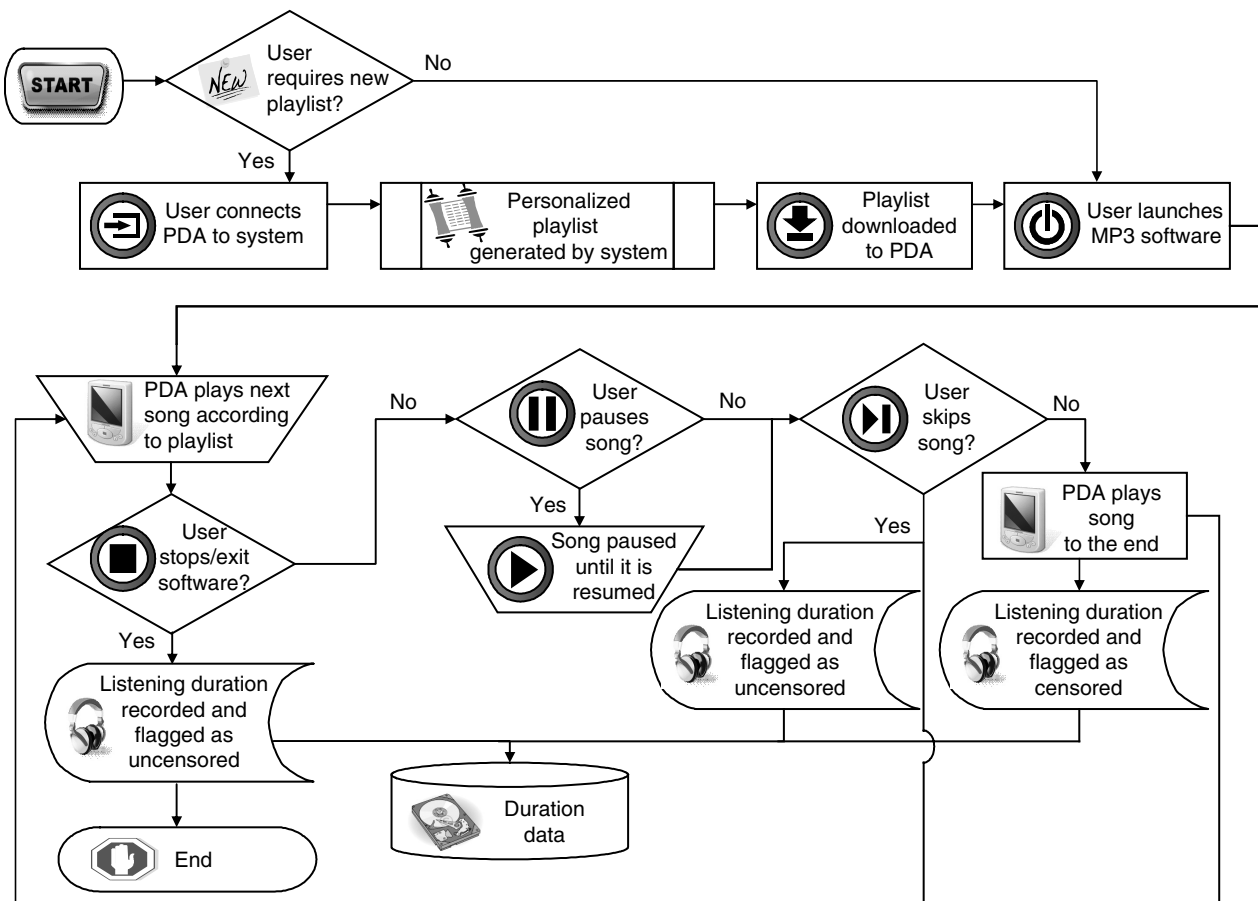**Figure 2   Handheld Palm TX PDA and Screen Shot of Controls**

Figure 3   User Interface Flow Chart of the Adaptive Personalization System

START → User requires new playlist? — No →

Yes ↓

User connects PDA to system → Personalized playlist generated by system → Playlist downloaded to PDA → User launches MP3 software

PDA plays next song according to playlist → User stops/exit software? — No →

User pauses song? — No → User skips song? — No → PDA plays song to the end

Yes ↓ Song paused until it is resumed

User stops/exit software? Yes ↓ Listening duration recorded and flagged as uncensored → End

Listening duration recorded and flagged as uncensored (User pauses Yes)

Listening duration recorded and flagged as censored (User skips Yes)

Duration data

4.2. Performance of the Adaptive Personalization System in Actual Use

The top left panel of Figure 4 shows how long (milliseconds) songs were listened to by the respondents, across the experimental and control groups and across the two waves. The distribution shows a spike at low values that quickly tapers off, which reveals that a large portion of songs was skipped early. Thus respondents seem to make a decision quickly which songs to listen to and which not. It is important to note that the first bar in the histogram does not present observations that are all exactly equal to zero, which would pose a problem for the lognormal hazard model that has no mass at zero. The first bar in the figure groups all observations from 0–10 seconds, the number of observations in the data that is exactly equal to zero is 2.02% only. These zeros are caused by (a) subjects stopping to listen before cycling through all songs, and (b) subjects accidentally tapping on the screen twice so that a song is skipped before it starts to play. We believe that this proportion of zeros is sufficiently small to be assumed negligible in the analysis, and we have added a small quantity (1 second) to these zeros before analysis. The distribution shows a second mode in the middle range for songs that were liked and listened to for much or all of their durations, indicating strong preferences for some of the songs. The top right panel of Figure 4 shows the distribution of the (posterior medians of the) parameter σ_i of the lognormal hazard function in both the first and second wave, across respondents and songs. The mode of the distribution is around 2.0 with very few values below 1.0, which indicates that a downward sloping hazard function generally describes respondents' listening behavior. Thus, the hazard function reflects that respondents do not need to listen to the songs for long before they can decide whether they like it or not. In addition, the distribution of the estimates indicates that for a large part of the respondents, the tendency to continue listening to a song increases as the listening duration increases.

The middle left- and right-hand panels of Figure 4 show the distribution of the number of nonzero parameters across respondents due to variable selection, for the song attributes and genres, respectively. The listening duration of over 25% of the respondents can be described using five song attributes or fewer, and duration of over 25% of the respondents is

Figure 4   Distribution of Key Parameters
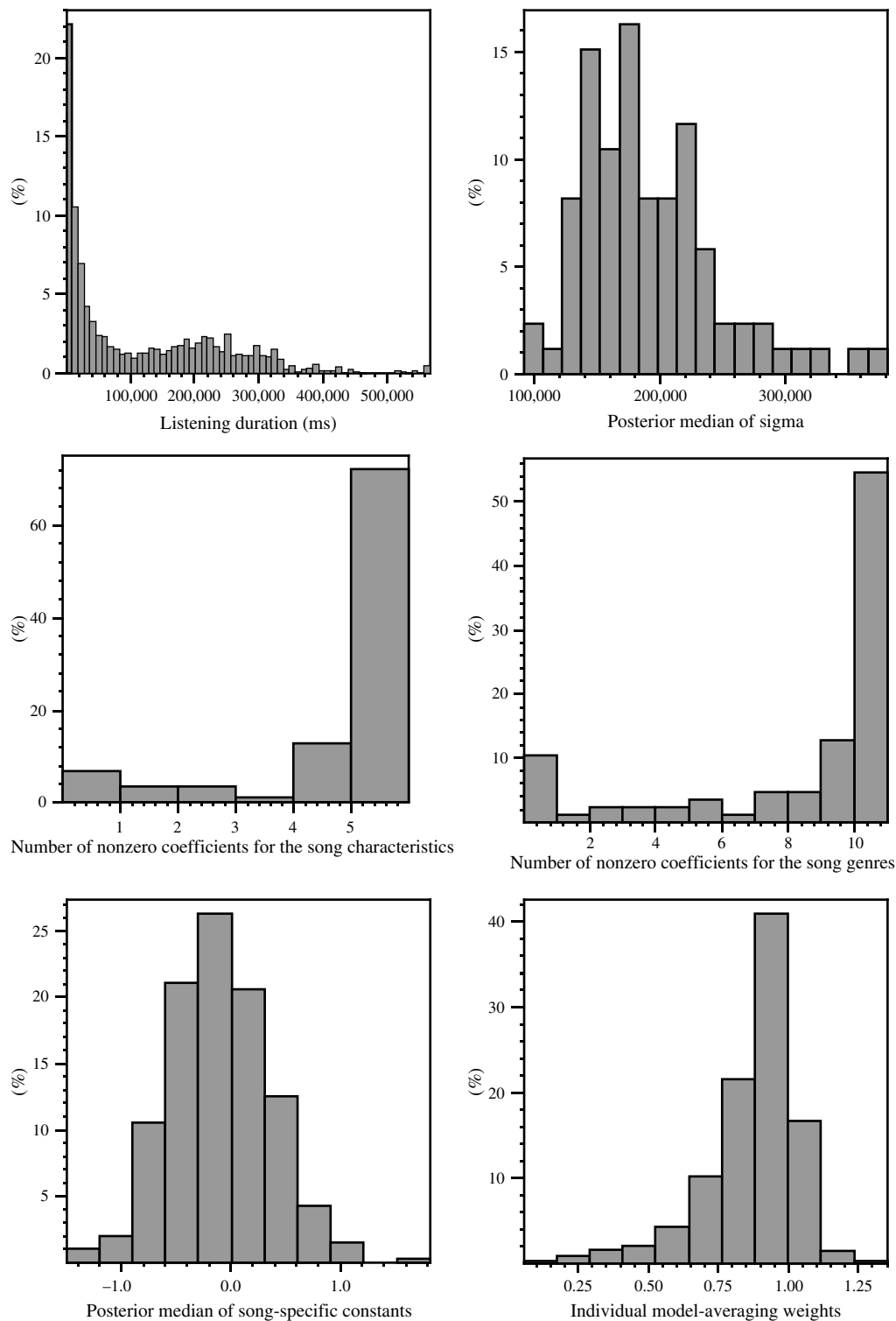
described with eight genres or fewer. For these individuals, the variable selection considerably simplifies the estimation. It appears that song tastes are concentrated among a relatively limited number of genres and attributes, which provides some evidence of noncompensatory preference formation.

The distribution of the (posterior median of) the song-specific constants of the aggregate model is shown in the bottom left panel of Figure 4. The distribution of these constants is relatively wide, because the songs that we used for our experiment represent a diverse collection of songs with idiosyncratic aspects

that may not be captured by song attributes and genres. The bottom right panel of Figure 4 shows the distribution of the between-individual model-averaging weights for all pairs of individuals. On average, a weight of about 0.87 is applied to other individual's parameters to predict the target individual's listening duration. However, there are instances where one individual's parameters are poor predictors of another individual's listening behavior (i.e., when the weight is 0.05). In the instances when the weights are above 1.00, the model-averaging approach helps to improve the predictions of the individuals' listening duration substantially. This happens in about 16% of the cases.

In our experiment the computation time using particle filtering was about 15% of that using MCMC on the same data. The fact that the data come in blocks of listening durations for the sequentially generated playlists helps to reduce the computation time of the particle filter. These gains will be even more pronounced when the data sets are larger than the relatively small data set in our experiment, which underscores the scalability of our algorithm.

Table 4 provides means and variances of the attributes for the songs in the database, as well as those in the playlists generated by the Adaptive Personalization System and the benchmark system that show significant differences in the second wave of the study. Among the attributes, "perceived loudness," "tempo," and, in particular, "dancing/listening" of the songs are significantly different between the playlists generated using the two methods. Songs with a higher value for "perceived loudness" and "tempo" have a louder volume and a faster tempo, respectively. A higher value for "dancing/listening" indicates that a song is more suitable for dancing, while a lower value indicates that a song is more suitable for easy listening. Thus on average, the songs

in the playlists generated by the benchmark method are more suitable for easy listening (and thus have softer volume and slower tempo) when compared to the songs in the playlists generated by the Adaptive Personalization System ($p < 0.05$). This difference in the song composition is reflected in how many of the songs fall into the "dance," "soul," and "vocal" song genres, with the proposed adaptive system generating playlists with more of the former and fewer of the latter two types of genres ($p < 0.05$). Since the benchmark method recommends songs with minimum dissimilarities and does not accommodate an element of "surprise" that prevents locking in on a narrow set of song-attribute values, the mean of this attribute is lower and its variance smaller, relative to the songs recommended by the Adaptive Personalization System. This indicates that the playlists generated by the benchmark are more homogeneously "easy listening" songs than those in the playlist for the Adaptive Personalization System. The proposed system generates playlists in which the songs are somewhat louder, have higher tempo, are more suited for dancing, and are more often orchestral in nature. In addition, testifying to the element of surprise in the playlists, the variance of these song attributes (i.e., nongenre attributes) tends to be somewhat higher as well.

In Table 5, we compare the performance of the personalization systems in terms of their effectiveness in recommending songs to individuals. We compute (1) the percentage of the duration of a song that was listened to by the respondents, and (2) the proportion of cases in which respondents listened to the entire song. Figure 5 presents the results graphically. When we compare the listening behavior of the respondents before and after we have customized the playlists (i.e., comparison of wave one and wave two), the playlists customized using our system results in a 25% improvement in the percentage of each song listened to, and a 33% improvement in the proportion of songs that were completely listened to ($p < 0.01$). Comparatively, the playlist customized using the benchmark approach results in only a 10% improvement in
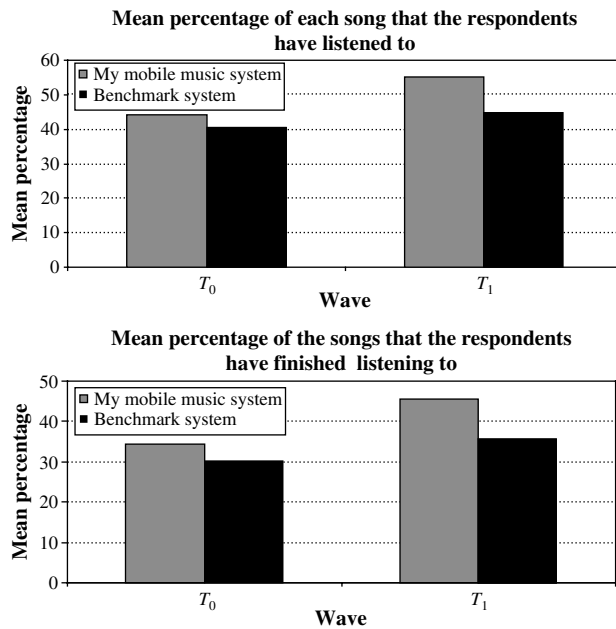
Table 4   Distribution of Several Song Characteristics in the Song Database and in the Personalized Playlists for the Adaptive Personalization System and the Benchmark

	Song attributes			Genres		
	Perceived loudness	Tempo	Listening/ dancing	Dance	Soul	Vocals
Mean						
Song database	0.99	0.99	0.87	0.14	0.05	0.03
Adaptive Personalization System	1.22	1.10	1.14	0.17	0.02	0.01
Minimum dissimilarity model	0.79	0.71	0.31	0.03	0.27	0.20
Variance						
Song database	1.00	1.00	0.98	0.12	0.05	0.03
Adaptive Personalization System	0.96	1.00	0.98	0.14	0.02	0.01
Minimum dissimilarity model	0.96	0.92	0.50	0.03	0.20	0.16

Table 5   Comparison of Adaptive Personalization System Performance to the Benchmark in the Pilot Implementation

	Mean percentage of each song that the respondents have listened to			Mean percentage of the songs that the respondents have finished listening to		
	T_0	T_1	ΔT	T_0	T_1	ΔT
Adaptive Personalization System	44.09	55.00	10.91	34.31	45.58	11.27
Minimum dissimilarity model	40.57	44.79	4.22	30.17	35.58	5.41

Figure 5   Comparison of Adaptive Personalization System Performance to the Minimum Dissimilarity Model in the Pilot Implementation

Mean percentage of each song that the respondents have listened to

Mean percentage of the songs that the respondents have finished listening to

the percentage of each song listened to and an 18% improvement in proportion of songs that were completely listened to ($p < 0.05$). In terms of improvement in the mean percentages from T_0 to T_1, our personalization system is 2.59 times ($=10.91/4.22$) as good as the minimum dissimilarity model in increasing the percentage of each song listened to, and is 2.08 ($=11.27/5.41$) times as good as the minimum dissimilarity model in increasing the number of songs that were completely listened to ($p < 0.01$).[10] Overall, the results on the actual listening behavior show that the Adaptive Personalization System is indeed superior to the benchmark and that it performs quite well in the pilot, which is illustrated in Figure 5.

5.   Discussion and Conclusions

The ability to collect customer information interactively over time makes it possible to implement

---

[10] We also asked for a subjective evaluation of the respondents' overall satisfaction with the songs after each wave of the experiment was completed (five-point scales). We analyzed the differences between the systems, controlling for gender, and found no significant difference in satisfaction with the songs between the systems (rank-order logistic regression: $\beta = 0.336$, standard error $= 0.405$, $p > 0.05$). Also, for neither of the two systems was the improvement significant over the initial random playlist. The absence of an improvement in satisfaction may be caused by respondents having the option to skip over songs they did not like, which happened much more often with the minimum dissimilarity model (Poisson regression: $\beta = 0.257$, standard error $= 0.0228$, $p < 0.01$). Another possible explanation is that personalization, although affecting individuals' listening behavior immediately, may take time to be consciously perceived.

Adaptive Personalization Systems that can provide increasingly targeted services to customers. This possibility is especially promising in the fast-growing information service sector. We develop such an Adaptive Personalization System and demonstrate it in the context of digital audio players.

Expansion of digital formats (e.g., MP3) and storage capabilities have made it extremely easy to amass huge collections of songs. Consumers often download thousands of songs into their mobile devices, many of which may not even be listened to. At the same time, advances in the portability of Internet access devices and services have greatly expanded the possibilities for real-time personalization. However, much of the potential for personalization remains to be tapped. In the present study, we have developed and implemented one of the first Adaptive Personalization Systems, applied here in the specific context of digital audio players. It personalizes playlists for mobile digital audio devices and enables automatic downloading of songs into these devices. The system consists of an adaptive personalization agent that analyzes individual-level listening behavior while it uses collaborative customization to further tailor playlists to individual tastes. At the same time, it promotes serendipity and circumvents locking in on a too-narrow set of user tastes. It requires little proactive user effort—important because the effort required for preference elicitation has an important influence on consumers' evaluation of the value of recommendation systems—and affects consumers' enjoyment in using them (Gretzel and Fesenmaier 2006). The primary targets for our Adaptive Personalization System are therefore consumers who value the convenience of an intelligent system that frees them from the effort of selecting which songs to download, of manually deciding which songs to play, and of rating songs as input to receiving song recommendations.

Although the Adaptive Personalization System involves little proactive effort from users, this generates some challenges as well, and there are several features that could be improved upon. First, similar to recommendation systems, our system suffers from a "cold start." Due to lack of new-user information we generate initial playlists at random. We have done this because we believe that asking for personal information may discourage the adoption of the personalization system early on. In addition, randomized initial playlists enable us to obtain better initial estimates of the model parameters, because song attributes in these playlists are uncorrelated with the model error by construction, and show more variation. However, we could alternatively ask users for demographic information and use that to initialize the system, ask users to initialize the system by manually selecting songs online after listening to snippets of them, or

ask them to allow the system to use playlists already stored on their devices for initialization.

Second, we have kept the MP3 software interface simple to reduce the effort needed to operate the mobile music-playing device. Users therefore do not to have the option to manually choose which songs or the order of songs to listen to. Some consumers may, however, want that ability to decide which songs to play next and may even enjoy the experience of searching for music. Obviously, that can be done in conjunction with using our system, and the controls in question can be relatively easily incorporated via the software of the mobile device. Some studies have indeed shown that giving users such control over the interaction with the recommendation system may result in more positive affective reactions (Xiao and Benbasat 2007).

Third, future research could extend the lognormal distribution assumed for the listening duration. In our applications only about 2% of the listening durations were zero, but in other applications this percentage may be larger, which is not captured well by the log-normal distribution. Accommodating the decision not to listen at all in the model may then improve fit and the generation of playlists. For example, a split-hazard model may be applied (Kamakura et al. 2004). We have not implemented this because of tractability and scalability of the algorithm. Alternatively, if zero listening durations are caused by individuals stopping the playlist, they may be treated as (ignorable) missing data (see Kamakura and Wedel 2000).

Fourth, to accommodate more rapidly changing music tastes, weighted updating could be considered for the particle filter, where waves of listening data further in the past receive less (or zero) weight, and therefore recent data contribute more to the parameter estimates. Alternatively, dynamic models could be developed that explicitly capture such changing preferences. Moreover, the effects of the context in which the user listens to the music and the sequence in which he or she listens to the songs could be addressed. For example, a user may have different music preferences when listening to music while at home, driving, working out, etc. Such effects may be incorporated in the model and exploited to further enhance the utility consumers get from the playlist. Different playlists can be generated for an individual depending on the listening context (car, gym, home, etc.). The system can either allow users to manually select the context or eventually automatically adapt to the context through the variable selection and particle filtering and select a context-specific playlist based on the observed listening behavior.

Finally, one may improve the attributes used to differentiate one song from the next, including composers and performing artists. Through a simulation study, Nikolaeva and Sriram (2006) show that more product attributes should be used in a recommendation system when the product is highly complex and when there is a large variability in tastes, which is the case for music. The Music Genome project can be used for increasing the number of attributes used to differentiate between songs. Several of these extensions will be explored in future research.

The Adaptive Personalization System that we have developed can be directly implemented because of its scalability accomplished through state-of-the-art computational methods, and through distributed processing accomplished through parallel computing on individuals' own mobile music devices. Our pilot implementation shows that the proposed Adaptive Personalization System leads to a greater number of songs listened to and songs listened to longer, while minimizing required user involvement in downloading and managing playlists. We expect this to result in (1) a stronger competitive position based on better meeting user needs and customer lock-in, and (2) greater volume of songs downloaded per user. Perceived personalization affects an individual's intention to adopt a recommendation system as it affects trust (Komiak and Benbasat 2006), and therefore our system may result in higher adoption and retention rates.

Although important, specific profit implications for companies that would adopt our system are difficult to provide without further research. We believe that the adoption of our system will increase the volume of downloads of songs by users, which would translate to greater revenue under the standard price-per-download model.[11] But the relationship between user listening behavior and download volume is as yet not established, and this is a topic for further research. However, the music industry has discussed moving to a different payment model for recorded music. Users could download digital music and keep it without any charges. Users would be encouraged to copy and share their music with others without restriction. The music industry would be paid each time a user listens to a song and payments would be collected by retailers who would sell music services. If such a payment model were adopted, our system would yield more than a 30% increase in revenue over the benchmark system, as shown in our pilot study. Alternatively, our system may have potential for implementation in (automated) music trading systems, where it may increase the volume of trades by better matching consumer tastes.

---

[11] Just to illustrate the benefits of personalization: Tam and Ho (2006) witness an increase in song download in a music website merely through a personalization of the Web contents.

We envision many other useful applications for Adaptive Personalization Systems, such as for movies, news items, and games.[12] However, these generalizations may require specific adaptations and extensions of the general procedure proposed here, which may be addressed in future research. Possible refinements notwithstanding, we believe that we have achieved our objectives of developing an operational and scalable Adaptive Personalization System that requires minimal proactive user effort, yet significantly increases the effectiveness of an important information service and performs much better than existing methods.

Acknowledgments

References

Aksoy, L., P. N. Bloom, N. H. Lurie, B. Cooil. 2006. Should recommendation agents think like people? *J. Service Res.* **8**(4) 297–315.

Ansari, A., K. Jedidi. 2000. Bayesian factor analysis for multilevel binary observations. *Psychometrika* **65**(4) 475–496.

Ansari, A., C. F. Mela. 2003. E-customization. *J. Marketing Res.* **40**(2) 131–145.

Ansari, A., S. Essegaier, R. Kohli. 2000. Internet recommendation systems. *J. Marketing Res.* **37**(3) 363–375.

Ariely, D., J. G. Lynch, Jr., M. Aparicio, IV. 2004. Learning by collaborative and individual-based recommendation agents. *J. Consumer Psych.* **14**(1&2) 81–95.

Arulampalam, S., S. Maskell, N. Gordon, T. Clapp. 2002. A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Processing* **50**(2) 174–188.

Bodapati, A. 2008. Recommender systems with purchase data. *J. Marketing Res.* **45**(1) 77–93.

Castelluccio, M. 2006. The music genome project. *Strategic Finance* **88**(6) 57–58.

Chipman, H., E. I. George, R. E. McCulloch. 2001. The practical implementation of Bayesian model selection. *Model Selection*, Vol. 38. Institute of Mathematical Statistics, Beachwood, OH, 65–116.

Cook, R. D., C. J. Nachtsheim. 1980. A comparison of algorithms for constructing exact *d*-optimal design. *Technometrics* **22**(3) 315–324.

Cooke, A. D. J., H. Sujan, M. Sujan, B. A. Weitz. 2002. Marketing the unfamiliar: The role of context and item-specific information in electronic agent recommendations. *J. Marketing Res.* **39**(4) 488–497.

Cosley, D., S. K. Lam, I. Albert, J. A. Konstan, J. Riedl. 2003. Is seeing believing? How recommender interfaces affect users' opinions. *Recommender Systems Soc. Comput.* **5**(1) 585–592.

Crisan, D., A. Doucet. 2002. A survey of convergence results on particle filtering methods for practitioners. *IEEE Trans. Signal Processing* **50**(3) 736–746.

Diehl, K. R., L. Kornish, J. G. Lynch, Jr. 2003. Smart agents: When lower search costs for quality information increase price sensitivity. *J. Consumer Res.* **30**(1) 56–71.

George, E. I., D. P. Foster. 2000. Calibration and empirical Bayes variable selection. *Biometrika* **87**(4) 731–747.

Gilbride, T. J., G. A. Allenby. 2006. A choice model with conjunctive, disjunctive, and compensatory screening rules. *Marketing Sci.* **23**(3) 391–406.

Gower, J. C. 1971. A general coefficient of similarity and some of its properties. *Biometrics* **27** 857–874.

Gretzel, U., D. R. Fesenmaier. 2006. Persuasion in recommender systems. *Internat. J. Electronic Commerce* **11**(2) 81–100.

Haaijer, R. E., M. Wedel, M. Vriens, T. J. Wansbeek. 1998. Utility covariances and context effects in conjoint MNP models. *Marketing Sci.* **17**(3) 236–252.

Häubl, G., K. B. Murray. 2003. Preference construction and persistence in digital marketplaces: The role of electronic recommendation agents. *J. Consumer Psych.* **13**(1–2) 75–91.

Häubl, G., V. Trifts. 2000. Consumer decision making in online shopping environments: The effects of interactive decision aids. *Marketing Sci.* **19**(1) 4–21.

Hoeting, J. A., D. Madigan, A. E. Raftery, C. T. Volinsky. 1999. Bayesian model averaging: A tutorial. *Statist. Sci.* **14**(4) 382–417.

Kamakura, W. A., M. Wedel. 2000. Factor analysis and missing data. *J. Marketing Res.* **37**(4) 490–498.

Kamakura, W. A., B. Kosslar, M. Wedel. 2004. Identifying innovators for the cross selling of new products. *Management Sci.* **50**(8) 1120–1132.

Komiak, S. Y. X., I. Benbasat. 2006. The effects of personalization and familiarity on trust and adoption of recommendation agents. *MIS Quart.* **30**(4) 941–960.

Kong, A., J. S. Liu, W. H. Wong. 1994. Sequential imputations and Bayesian missing data problems. *J. Amer. Statist. Assoc.* **89**(425) 278–288.

Kuang, C. 2007. Free music now! Lala.com's plan to give songs away could upend the industry. *Wired* (October 23), http://www.wired.com/entertainment/music/magazine/15-11/ff_lala.

Kuhfeld, W. F., R. D. Tobias, M. Garratt. 1994. Efficient experimental design with marketing research applications. *J. Marketing Res.* **31**(4) 545–557.

Lam, S. K., J. Riedl. 2004. Shilling recommender systems for fun and profit. *WWW '04: Proc. 13th Internat. Conf. World Wide Web*, ACM Press, New York, 393–402.

Madigan, D., A. E. Raftery. 1994. Model selection and accounting for model uncertainty in graphical models using Occam's window. *J. Amer. Statist. Assoc.* **89** 1535–1546.

Montgomery, A. L., K. Hosanagar, R. Krishnan, K. B. Clay. 2004. Designing a better shopbot. *Management Sci.* **50**(2) 189–206.

Nikolaeva, R., S. Sriram. 2006. The moderating role of consumer and product characteristics on the value of customized on-line recommendation. *Internat. J. Electronic Commerce* **11**(2) 101–123.

Ogungbenro, K., G. Graham, I. Gueorguieva, L. Aarons. 2005. The use of a modified Federov exchange algorithm to optimise sampling times for population pharmacokinetic experiments. *Comput. Methods Programs Biomedicine* **80** 115–125.

Pronzato, L., É. Thierry. 2003. Sequential experimental design and response optimisation. *Statist. Methods Appl.* **11** 277–292.

Rainie, L. M. Madden. 2005. Podcasting. Data memo, Pew Internet & American Life Project, Washington, D.C., http://www.pewinternet.org/pdfs/PIP_podcasting2005.pdf.

---

[12] The TIVO movie recommendation system also employs a hybrid system which is scalable. However, it requires explicit inputs from the viewers in the form of show ratings. Implicit ratings are obtained by inferring that a user who records a movie prefers that movie. The TIVO algorithm is conceptually similar to our benchmark model.

Randall, T., C. Terwiesch, K. T. Ulrich. 2007. User design of customized products. *Marketing Sci.* **26**(2) 268–280.

Ridgeway, G., D. Madigan. 2003. A sequential Monte Carlo method for Bayesian analysis of massive data sets. *Data Mining Knowledge Discovery* **7**(3) 301–319.

Rossi, P., G. Allenby. 2003. Bayesian statistics and marketing. *Marketing Sci.* **22**(3) 304–328.

Rossi, P. E., G. Allenby, R. McCulloch. 2005. *Bayesian Statistics and Marketing*. John Wiley & Sons, Chichester, West Sussex, UK.

Rossi, P. E., Z. Gilula, G. M. Allenby. 2001. Overcoming scale usage heterogeneity: A Bayesian hierarchical approach. *J. Amer. Statist. Assoc.* **96**(453) 20–31.

Rust, R. T., T. S. Chung. 2006. Marketing models of service and relationships. *Marketing Sci.* **25**(6) 560–580.

Saha, A., L. Hilton. 1997. Expo-power: A flexible hazard function for duration data models. *Econom. Lett.* **54** 227–233.

Sandor, Z., M. Wedel. 2001. Designing conjoint choice experiments using managers' prior beliefs. *J. Marketing Res.* **38**(4) 430–444.

Sha, N., M. G. Tadesse, M. Vannucci. 2006. Bayesian variable selection for the analysis of microarray data with censored outcomes. *BioInformatics* **22**(18) 2262–2268.

Sun, B. 2006. Technology innovation and implications for customer relationship management. *Marketing Sci.* **25**(6) 594–597.

Syam, N. B., N. Kumar. 2006. On customized goods, standard goods, and competition. *Marketing Sci.* **25**(5) 525–537.

Tam, K. Y., S. Y. Ho. 2006. Understanding the impact of web personalization on user information processing and decision outcomes. *MIS Quart.* **30**(4) 865–890.

Verdinelli, I., J. B. Kadane. 1992. Bayesian designs for maximizing information and outcome. *J. Amer. Statist. Assoc.* **87**(418) 510–515.

Xiao, B., I. Benbasat. 2007. E-commerce product recommendation agents: Use, characteristics, and impact. *MIS Quart.* **31**(1) 137–209.

Ying, Y. P., F. Feinberg, M. Wedel. 2006. Leveraging missing ratings to improve online recommendation systems. *J. Marketing Res.* **43**(3) 355–365.