Supporting Context-Aware Media Recommendations for Smart Phones

A context-aware media recommendation platform uses an N × M-dimensional model and a hybrid processing approach to support media recommendation, adaptation, and delivery for smart phones.

Zhiwen Yu and Xingshe Zhou
Northwestern Polytechnical University, P.R. China

Daqing Zhang
Institute for Infocomm Research, Singapore

Chung-Yau Chin, Xiaohang Wang, and Ji Men
National University of Singapore

Equipped with network connectivity such as Bluetooth or a General Packet Radio Service, smart phones can access media content in a variety of formats—video, audio, image, or text. This precipitates the need to recommend the right content in the right form to the right person.

Traditional recommender systems provide recommendations based only on user preference, using methods that we can classify as collaborative, content-, demographic-, knowledge-, or utility-based. To improve performance, researchers sometimes combine these methods[1–3] or incorporate situational context information, such as location and time.[4] For more personalized multimedia delivery, some researchers have considered both user preference and the device or network context to determine the appropriate presentation method.[5–7] However, we have yet to see a recommendation system that can handle all three context categories—*user preference*, *situation context*, and *capability context*. Furthermore, recommendation systems often consider only a particular application's specific context information; they seldom propose a formal, effective recommendation model or algorithms for dealing with a wide variety of context information.

To provide media recommendations for smart phones based on all three context categories, we present a generic and flexible $N \times M$-dimensional (N2M) recommendation model. The model considers context information—ranging from user preference and situation to device and network capability—as input for both content and presentation recommendations. Furthermore, it generates multidimensional output, offering a content rating and recommendation. Because a single recommendation method can't deal with all three context categories, we propose a hybrid recommendation approach to synergize content-based, Bayesian-classifier, and rule-based methods. Based on the recommendation model and hybrid-processing approach, we built a *co*ntext-aware *me*dia *r*ecommendation platform called CoMeR to support media recommendation, adaptation, and delivery for smart phones.

The description model

To ensure interoperability with third-party services and applications, we adopted the MPEG-7 description schema (DS) to represent media metadata, and we developed a context representation model using OWL (Web Ontology Language).[8]

Media description

We use the MPEG-7 Creation DS and Classification DS to describe information about the media item, such as title, keyword, direc-

Published by the IEEE CS and IEEE ComSoc ■ 1536-1268/06/$20.00 © 2006 IEEE

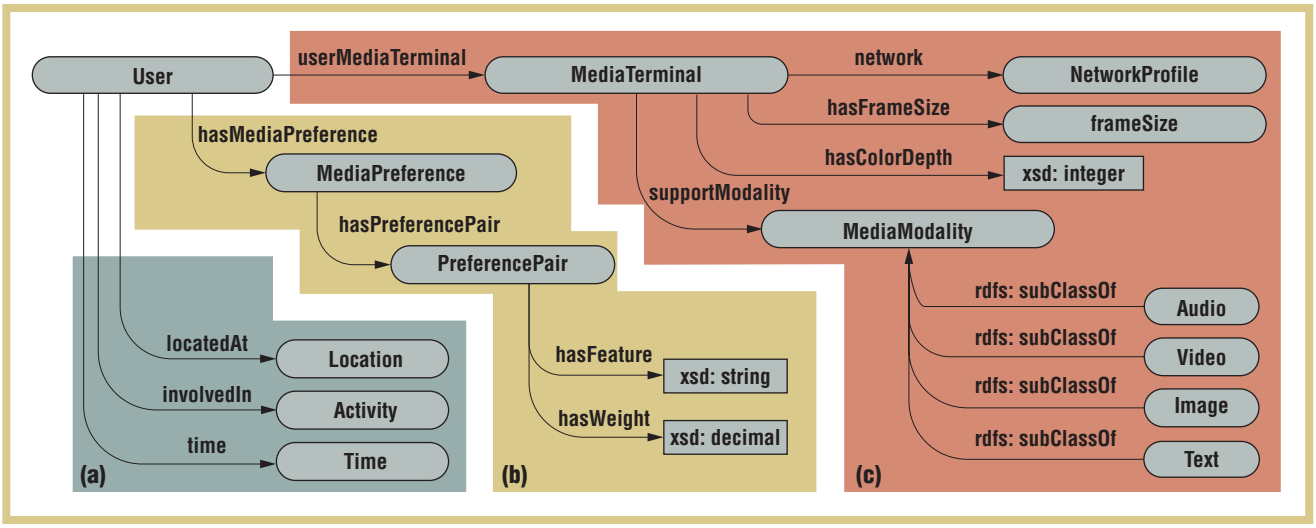


Figure 1. A partial context ontology, which delineates the (a) user situation context, (b) user media preference, and (c) media terminal's capability.

tor, actor, genre, and language. Our system then uses this information to match user preferences. The Variation DS specifies the media content's various formats and variation relationships (that is, how the variation is built from the source), letting CoMeR select the most appropriate format on the basis of the terminal device's capabilities and the network conditions.

Context description

We used an ontology-based context model for context representation.[9] This model adopts OWL as the representation language to enable expressive context description and data interoperability with third-party services and applications. In the knowledge representation domain, the term "ontology" refers to the formal and explicit description of domain concepts, which are often conceived as a set of entities, relations, instances, functions, and axioms.

Figure 1 shows a partial context ontology, delineating the user's situational context and media preferences and the media terminal's capability. Our system captures and evaluates the user's operating context in terms of his or her location, the activity, and the time. The MediaPreference class denotes a user preference regarding the media content by indicating the <*feature, weight*> preference pair. Weight, ranging from –1 to 1, indicates the user's preference level for the corresponding feature. The MediaTerminal class refers to the device's operating capacity—its display characteristics, network communication profile, and supported media modality.

The recommendation model

The N2M model provides multimedia recommendations over multiple dimensions to generate multidimensional output. We define user preference, terminal capability, location, time, and so on as context dimensions. We define format, frame rate, frame size, score (similar to rating), and so on as quality-of-service dimensions, which constitute the recommendation output.

If CD_1, CD_2, …, CD_{N-1} are context dimensions, and QD_1, QD_2, …, QD_M are output QoS dimensions, then we can define the recommendation model as

$$R: MediaItem \times CD_1 \times CD_2 \times \ldots \times CD_{N-1} \rightarrow QD_1 \times QD_2 \times \ldots \times QD_M$$

The input is an N-dimensional space comprising one dimension of *MediaItem* and $N - 1$ dimensions of context. The output is an M-dimensional space including M QoS dimensions.

Figure 2 illustrates the recommendation, which takes 3D multimedia input and recommends a 2D multimedia output. In other words, the recommendation function is

$$R(MediaItem, UserPreference, Terminal\ Capability) = (Modality, Score).$$

The input includes the following dimensions:

- *MediaItem* enumerates all the multimedia items that the system can recommend—each one defined as a vector of description features (title, genres, actors, and keywords).
- *UserPreference* represents all users' preferences—each one represented as a vector of <feature, weight> pairs.
- *TerminalCapability* refers to all combinations of displaying capability: video, image, and text—such as (Video+Image+Text) or (Image+Text).

For the output,

- *Modality* represents the final recommended format for a multimedia item—video, image, or text, and

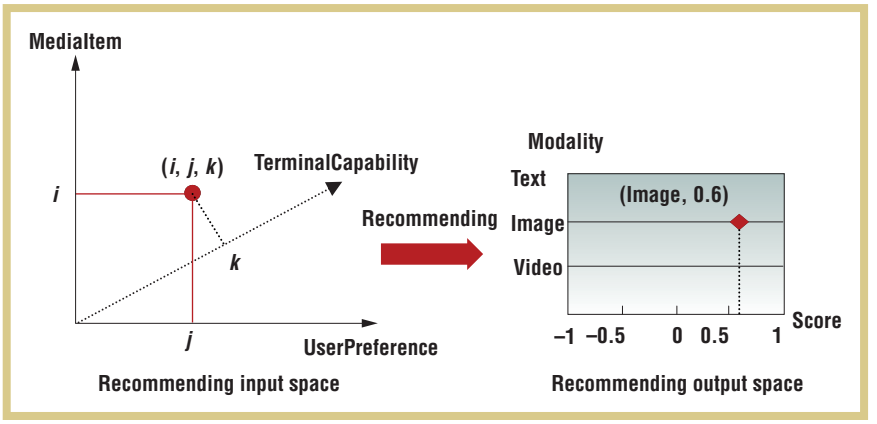


Figure 2. Our system takes 3D input (*MediaItem* × *UserPreference* × *TerminalCapability*) and recommends 2D output (*Modality* × *Score*).

- *Score* represents the degree of user interest in the recommended item, ranging from –1 (least preferred) to 1 (most preferred).

In figure 2, we assume

- *MediaItem*(*i*) = ((Titanic), (Drama, Romance), (Leonardo DiCaprio, Kate Winslet), (love, diamond, disaster)),
- *UserPreference*(*j*) = ((Romance, 0.70), (Love, 0.5)), and
- *TerminalCapability*(*k*) = (Image + Text).

The output, *R* = (Image, 0.6), suggests that the item should be presented as an image, and the user's interest in the item is 0.6. The score depends on preference and situation context, while the presenting form is determined by capability context, such as whether the terminal can play video and in which format.

Hybrid recommendation approach

To generate the recommendation just presented, we deployed three different approaches and then combined them to create our hybrid approach.

Content-based approach

We adopted a content-based recommendation method to evaluate media items against preference context. Content-based recommendation compares features in a media item with terms that characterize user preference to determine whether the user likes that item. Only media items with a high degree of similarity to the user's preference are recommended.

We used the Vector Space Model to represent object information.[10] We define the user profile as a vector $\mathbf{P} = (w_1, ..., w_n)$, where w_i is the weight of term t_i in the profile. Similarly, a media item can be also represented as a vector with n elements, $\mathbf{C} = (u_1, ..., u_n)$, where u_i is the weight assigned to term t_i, which is the same as that in the user profile vector. Because all terms aren't equally important for media representation—for instance, terms in the Actor field might be more important than those in the Keyword field—we assign importance factors to reflect each term's relative importance for media item identification. We define a field set, S = {Genre, Actor, Keyword}, and an importance factor set, $W = \{W_x \mid x \in S\}$. W_x is the importance factor assigned to terms in field x to reflect their relative importance.

Based on these definitions, the system assigns the weight u_i in the media item vector, complying with the following rules:

1. If term t_i is merely included in field x of the media item's metadata, then $u_i = W_x$;
2. If t_i is included in two or more fields, then $u_i = Max\{W_x\}$ (the maximum value of W_x that t_i is included in corresponding fields);
3. If t_i isn't included in any field, then $u_i = 0$.

In the classical vector space representation, the similarity between the two vectors of the media item and user profile indicates the degree of relevance between the media and user interests. We adopt the commonly used similarity metric—the cosine value of the angle between the two vectors—as the evaluating measure between the media item and preference context. Given a media item $\mathbf{C} = (u_1, ..., u_n)$ and a user profile $\mathbf{P} = (w_1, ..., w_n)$, the cosine similarity is calculated as

$$Similarity(\mathbf{C},\mathbf{P}) = \frac{\mathbf{C}\cdot\mathbf{P}}{\|\mathbf{C}\|\times\|\mathbf{P}\|} = \frac{\sum_{i=1}^{n} u_i w_i}{\sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\sum_{i=1}^{n} w_i^2}}$$

Figure 3 illustrates the process of feature extraction and similarity measurement (assuming $W_{Genre} = 0.75$, $W_{Actor} = 0.50$, $W_{Keyword} = 0.25$). The media item is *I, Robot*, and the similarity calculated is 0.75.

Bayesian-classifier approach

We adopted the Naïve Bayes classifier approach to evaluate media items against the situation context.[11] We grouped each situation context dimension's values into classes. For example, we can divide a user's location into three classes—Home, Office, or Outdoors (on a bus, for example)—and divide a user's viewing time into two classes—Weekday or Weekend. We evaluate the probability of a media item belonging to a certain class of a context or a combination of context classes. For example, what is the probability of the user viewing the movie *Gone with the Wind* on a weekend—**P**(*Weekend*|*Gone with the Wind*)? Suppose $\mathbf{C}_1, \mathbf{C}_2, ..., \mathbf{C}_j, ..., \mathbf{C}_k$ are k classes of a considered situation

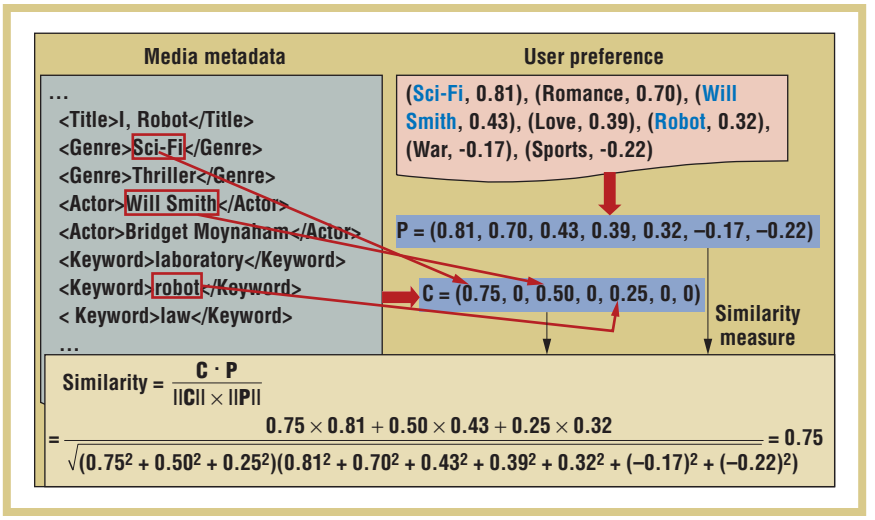


Figure 3. An example of feature extraction and similarity measurement. User preference is extracted and represented as P; the media item is extracted and represented as C; then the similarity is calculated.

context. If time is considered in the recommendation, we set C_1 = Weekday and C_2 = Weekend. Then, we can calculate the probability of media item $\vec{x}$ belonging to class C_j through statistical analysis of the user's viewing history:

$$P(C_j \mid \vec{x}) = P(\vec{x} \mid C_j)P(C_j) / P(\vec{x})$$

$$P(\vec{x} \mid C_j) = \prod P(f_i \mid C_j)$$

$$P(C_j) = k(C_j) / T$$

$$P(\vec{x}) = \sum_{j=1}^{k} P(\vec{x} \mid C_j)P(C_j)$$

$$P(f_i \mid C_j) = \frac{n(f_i, C_j) + 0.5}{n(C_j) + 0.5|V_j|}$$

where

- f_i denotes the ith feature of $\vec{x}$,
- $k(C_j)$ denotes the number of viewing records in C_j,
- T denotes the total record number in the viewing history,
- $n(f_i, C_j)$ denotes the times f_i occurs in C_j,
- $n(C_j)$ denotes the sum of times all features occur in C_j, and
- $|V_j|$ denotes the total number of features occurring in C_j.

Given a class C_j (a particular context value, such as Weekend), the system would recommend only the media items that have a high degree of $P(C_j \mid \vec{x})$.

For example, we can estimate the probability that a user will view *I, Robot* outdoors on a weekend. Because we're considering both location and time, we set C_1 = (Home, Weekday), C_2 = (Home, Weekend), C_3 = (Office, Weekday), C_4 = (Office, Weekend), C_5 = (Outdoors, Weekday), and C_6 = (Outdoors, Weekend). The general form of a record in the user's viewing history is

<location|time|title|genre|actor|keyword>

Location and **time** determine where and when the viewing action takes place, respectively; **title** is the name of the media item viewed; and **genre**, **actor**, and **keyword** are features of the media item. Figure 4 shows a sample viewing history. According to the equations we presented and figure 4's viewing history, $P(C_6 \mid I, Robot) = P(I, Robot \mid C_6)P(C_6)/P(I, Robot) = 0.38$.

Figure 4. A sample viewing history.

Rule-based approach

The recommended item's modality, format, frame rate, frame size, and so forth should satisfy the capability context. We adopted a rule-based approach to evaluate media items against capability context. This approach determines which choice should be taken in a specific situation, using a set of situation-result rules. Each rule is an if-then clause in nature. In our system, the rule-based approach infers the presenting form from the capability context.

Our current system applies the Jena2 generic rule engine to support forward-chaining reasoning over the context.[12] The rule engine is responsible for interpreting rules, connecting to the OWL-represented capability context, and eliminating conflicts.

For example, the rules in figure 5 infer the presenting modality based on terminal capability (the modality supported—video, image, or text) and network condition (bandwidth). The third rule infers the bandwidth level as Middle when the available bandwidth is between 4 and 50 Kbps. The fifth rule specifies that even if the terminal device can display video, image, and text, if the available bandwidth is Middle, the suggested modality is the image format.

Synergizing the approaches

We synergized the content-based,

1. type(?bandwidth, Bandwidth), greaterThan(?bandwidth, 50Kbps) =>bandwidthLevel(?bandwidth, High)
2. type(?bandwidth, Bandwidth), lessThan(?bandwidth, 4Kbps) =>bandwidthLevel(?bandwidth, Low)
3. type(?bandwidth, Bandwidth), lessThan(?bandwidth, 50Kbps), greaterThan(?bandwidth, 4Kbps) =>bandwidthLevel(?bandwidth, Middle)
4. type(?capability, Capability), type(?bandwidth, Bandwidth), type(?modality, Modality), modalitySupported(?capability, Video+Image+Text), bandwidthLevel(?bandwidth, High) =>modalityRecommended(?modality, Video)
5. type(?capability, Capability), type(?bandwidth, Bandwidth), type(?modality, Modality), modalitySupported(?capability, Video+Image+Text), bandwidthLevel(?bandwidth, Middle) =>modalityRecommended(?modality, Image)
6. type(?capability, Capability), type(?bandwidth, Bandwidth), type(?modality, Modality), modalitySupported(?capability, Video+Image+Text), bandwidthLevel(?bandwidth, Low) =>modalityRecommended(?modality, Text)
7. type(?capability, Capability), type(?bandwidth, Bandwidth), type(?modality, Modality), modalitySupported(?capability, Image+Text), bandwidthLevel(?bandwidth, High) =>modalityRecommended(?modality, Image)
8. type(?capability, Capability), type(?bandwidth, Bandwidth), type(?modality, Modality), modalitySupported(?capability, Image+Text), bandwidthLevel(?bandwidth, Low) =>modalityRecommended(?modality, Text)

Figure 5. Sample rules that infer appropriate modality based on the terminal capabilities and the network conditions.

Bayesian-classifier, and rule-based approaches to generate recommendations as follows.

First, we used the content-based approach to measure the similarity between a media item and the preference context. Then, we used the Naïve Bayes classifier approach to calculate the probability of the item belonging to the situation context, $\mathbf{P}(\mathbf{C}_j|\vec{x})$. Finally, we used a weighted linear combination of these two subscores to get the overall score,

$$Score = W_p \times Similarity + W_s \times \mathbf{P}(\mathbf{C}_j|\vec{x})$$

where W_p and W_s are weighting factors ($W_p + W_s = 1$; $0 \le W_p \le 1$; and $0 \le W_s \le 1$) reflecting the relative importance of preference and situation context.

If we only consider the preference context, then $W_p = 1$ and $W_s = 0$. However, if we only consider the situation context, then $W_p = 0$ and $W_s = 1$. If we consider both the preference and situation context, W_p and W_s are assigned two positive real numbers—for example, $W_p = 0.6$ and $W_s = 0.4$. Therefore, for the previously mentioned media item, *I, Robot*, the final recommendation score is $0.6 \times 0.75 + 0.4 \times 0.38 = 0.602$.

We rank all media items according to the scores achieved through these three steps and choose the highest score or three highest-scored items for a user's choices. Then, we use the rule-based approach to determine the appropriate form of the item to be presented, given the capability context.

The CoMeR architecture

Using the N2M recommendation model and hybrid processing approach, we built the CoMeR platform to support context-aware media recommendation for smart phones.

The CoMeR platform leverages our Semantic Space context infrastructure[9] for infrastructure-based systematic context acquisition. Semantic Space exploits Semantic Web technologies to support explicit representation, expressive querying, and flexible reasoning of contexts in smart spaces. CoMeR acquires necessary context information of the three categories from the context knowledge base through the context query engine.

CoMeR comprises five collaborating components: a media content database, media metadata database, recommendation engine, media adapter, and media deliverer (see figure 6). The media content database stores media of various kinds of media modalities, file formats, bit rates, and resolutions. The Xindice XML database (see http://xml.apache.org/xindice), which has efficient querying capabilities, stores the media metadata (MPEG-7 XML files).

The recommendation engine estimates a media item's score and determines its presentation form, given the context. It queries the context knowledge base for context information and the metadata base for the media description and then recommends a media item (or the top three). If the selected item has already been prepared with the variation in an appropriate form (described in the Variation DS part of the multimedia metadata), it invokes the media deliverer directly; otherwise, it first calls the media adapter to make the adaptation and then triggers the media deliverer.

The media adapter must adapt the content on the basis of the terminal's capability and the network bandwidth. It transforms the content from one media type to another so a particular device can process the content and a specific network condition can deliver it. Systems can adapt the media adaptation statically at authoring time, prior to delivery, or dynamically, on-the-fly if needed. CoMeR's media adapter currently prepares the content variations before delivery.

The media deliverer is responsible for streaming or downloading media content to various smart phones through different networks. If the recommended modality is video or audio, the media deliverer streams the content to terminals. On the other hand, if the modality is image or text, the media deliverer just downloads the content. The media deliverer supports a wide variety of video, audio, and image formats and can adapt to different terminals and network conditions. CoMeR integrates the Java

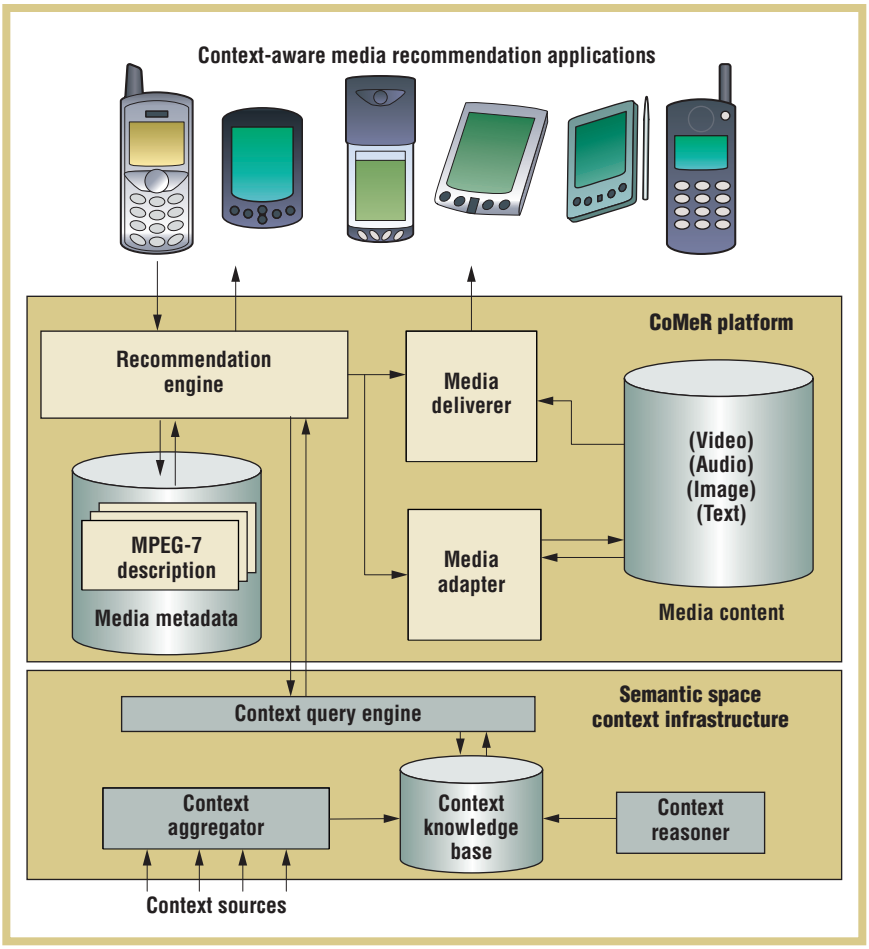


Figure 6. Our context-aware media recommendation (CoMeR) platform's architecture.

Media Framework and Windows Media Server for media streaming.

CoMeR supports various context-aware media recommendation applications by providing developers with APIs to access its functionalities. Application developers need only specify which context information to consider, and which QoS criteria will be composed as output in the recommendation. They need not care how the recommendation, adaptation, and delivery are accomplished.

Moreover, we adopted a component-oriented approach when building CoMeR. We developed the recommendation algorithms as software components, so the CoMeR architecture is pluggable and scalable. It supports efficient life-cycle management of components, ranging from install, start, remove, and upgrade to stop actions.

Implementation and evaluation

We implemented the hybrid recommendation approach and integrated it into the CoMeR platform, which we developed with Java. For scalability, we adopted the component-based programming model—Enterprise JavaBeans—in the implementation.

Based on CoMeR, we developed a context-aware movie recommender for smart phones called ContAwareMovie. We deployed the CoMeR platform and the Semantic Space context infrastructure on a workstation acting as a media server. The client-side application ran on a Sony Ericsson P900 smart phone. When developing the application, we followed the Mobile Information Device Profile provided by J2ME (http://java.sun.com/j2me). The media server recommends movies on the basis of the context information.

Using user preference, location, and time as input context, we evaluated the movie items. As a result, *I, Robot* scored the highest at 0.602. As network conditions changed, the modalities presented varied accordingly. For example, with high network bandwidth (128.4 Kbps), the content was displayed as video (see figure 7a), and with very low bandwidth, the content was displayed as text (see figure 7b). Figure 7c shows the context information for the low bandwidth connection.

From January to March 2005, we invited nine users (postgraduate students in the Institute for Infocomm Research lab) to use and evaluate CoMeR. The data sets used for performance analysis were taken from the Internet Movie Database (http://us.imdb.com).

We mainly evaluated our system by measuring the time it spent on recommendations. The time factor is crucial, because long delays create negative user experiences. We measured the recommending time on a Dell workstation with a 2.4-GHz Pentium 4 CPU, and 512 Mbytes of RAM running Windows XP by varying the number of movie metadata (also movies) from 40 to 200. When a user searched for a movie, CoMeR launched the recommendation engine, which calculated the similarity between the user preference and each movie's metadata, determined the probability of each movie viewed in a specific location and time, and then inferred the displaying modality of the selected content. The green line in figure 8 shows the initial result. The recommendation process was computationally intensive—the recommendation time increased proportionally according to the movie database's size.

If the movie database becomes very large, the system won't scale well. To handle scalability, we split computa-

Figure 7. A smart phone using the CoMeR platform to determine when to display (a) video (during high network bandwidth) and (b) text (during very low bandwidth). The phone also displays (c) context information about the low bandwidth (3.7 Kbps).

tions associated with the recommendation into offline and online phases, similar to Eigentaste.[13] The offline phase measured the similarity and determined the probability beforehand. The online phase, upon a user's recommendation request, inferred the displaying modality of the selected content. Because the offline computation was always running in the background, and its time was transparent to the user, only the online time was crucial for the performance evaluation. The red line in figure 8 shows the improved result (approximately 0.3 seconds), which the test participants found to be acceptable.

We can further deal with large vectors of ratings and dimensionality reduction by adopting techniques such as neighbor-based collaborative filtering in GroupLens[14] and singular value decomposition in Latent Semantic Analysis.[15]

From the user's perspective, we also evaluated the system's filtering effectiveness. It indicated the quality of recommendation—that is, whether the system could recommend the right content to the user. We calculated *precision* and *recall*[16] values for each user. Precision is the ratio of the amount of recommended content in which the user was interested to the amount of content recommended, while recall is the ratio of the amount of recommended content of interest to the amount of content of interest in the collection:

$$\text{Precision} = \frac{recommended \cap interested}{recommended}$$

$$\text{Recall} = \frac{recommended \cap interested}{interested}$$

The result was encouraging, with most precision values around 0.8 and recall values ranging from 0.63 to 0.75.

Incorporating contextual information into the recommendation process is crucial for ubiquitous media recommendations. In the future, we plan to investigate and incorporate dynamic adaptation into the media adapter to generate media variations on-the-fly according to the available network bandwidth. We also plan to improve system trustworthiness in terms of security, privacy, and usability. To provide a better user experience, we plan to enlarge the movie database and integrate automatic user preference learning. P

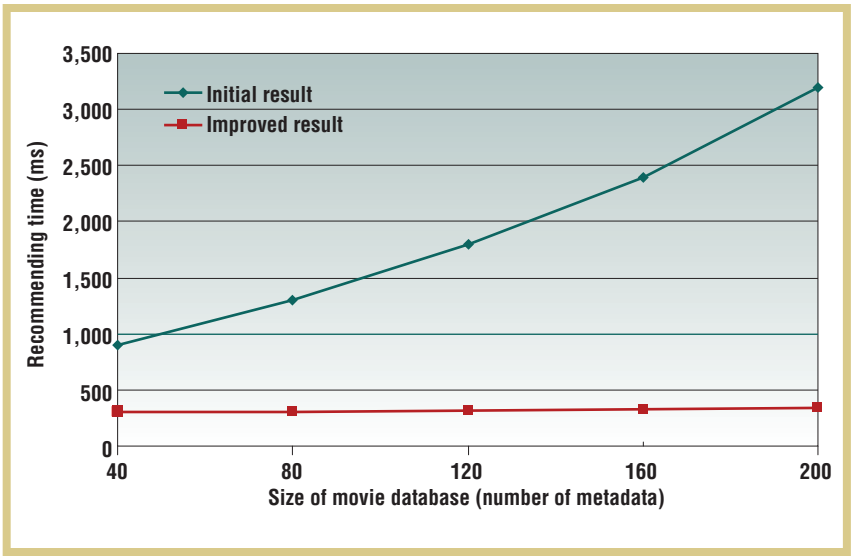


Figure 8. Time CoMeR spent making recommendations. We addressed scalability by splitting computations associated with the recommendation into offline and online phases, improving CoMeR's performance (red line).

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable comments and suggestions. This work was supported by the National Science Foundation of China and the Doctorate Foundation of Northwestern Polytechnical University of China.

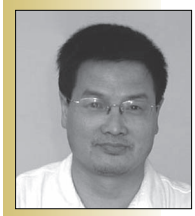
REFERENCES

1. R. Burke, "Hybrid Recommender Systems: Survey and Experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, 2002, pp. 331–370.
2. M. Pazzani, "A Framework for Collaborative, Content-Based and Demographic Filtering," *Artificial Intelligence Rev.*, vol. 13, no. 5, 1999, pp. 393–408.
3. M. Balabanovic and Y. Shoham, "Fab: Content-Based, Collaborative Recommendation," *Comm. ACM*, vol. 40, no. 3, 1997, pp. 66–72.
4. G. Adomavicius et al., "Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach," *ACM Trans. Information Systems*, vol. 23, no. 1, 2005, pp. 103–145.
5. B.L. Tseng et al., "Using MPEG-7 and MPEG-21 for Personalizing Video," *IEEE MultiMedia*, vol. 11, no. 1, 2004, pp. 42–52.
6. O. Steiger et al., "MPEG-Based Personalized Content Delivery," *IEEE Int'l Conf. Image Processing* (ICIP 03), IEEE CS Press, 2003, pp. 45–48.
7. T. Lemlouma and N. Layaida, "Encoding Multimedia Presentations for User Preferences and Limited Environments," *Proc. IEEE Int'l Conf. Multimedia and Expo* (ICME 03), IEEE CS Press, 2003, pp. 165–168.
8. D.L. McGuinness and F. van Harmelen, "OWL Web Ontology Language Overview," *W3C Recommendation*, 2004; www.w3.org/TR/owl-features.
9. X. Wang et al., "Semantic Space: An Infrastructure for Smart Spaces," *IEEE Pervasive Computing*, vol. 3, no. 3, 2004, pp. 32–39.
10. G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*, Addison-Wesley, 1989.
11. T. Joachims, "A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization," *Proc. 14th Int'l Conf. Machine Learning*, Morgan Kaufmann, 1997, pp. 143–151.
12. J.J. Carroll et al., "Jena: Implementing the Semantic Web Recommendations," *Proc. 13th Int'l Conf. World Wide Web*, ACM Press, 2004, pp. 74–83.
13. K. Goldberg et al., "Eigentaste: A Constant Time Collaborative Filtering Algorithm," *Information Retrieval*, vol. 4, no. 1, 2001, pp. 133–151.
14. J. Herlocker et al., "An Algorithmic Framework for Performing Collaborative Filtering," *Proc. 22nd Ann. Int'l Conf. Research and Development in Information Retrieval* (SIGIR99), ACM Press, 1999, pp. 230–237.
15. S. Deerwester et al., "Indexing by Latent Semantic Analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, 1990, pp. 391–407.
16. C.J. Rijsbergen, *Information Retrieval*, 2nd ed., Butterworths, 1979.

the AUTHORS

Zhiwen Yu is a postdoctoral researcher at Nagoya University, Japan. His research interests include pervasive computing, context-aware systems, intelligent information technology, and personalization. He received his PhD in computer science from the Northwestern Polytechnical University, P.R. China. The work reported on in this article was done when he was a PhD candidate at the School of Computer Science, Northwestern Polytechnical University. He's a member of the IEEE. Contact him at the Information Technology Center, Nagoya Univ., Furo-cho, Chikusa-ku, Nagoya, Japan, 464-8601; zhiwen@itc.Nagoya-u.ac.jp.

Daqing Zhang is manager of the Context-Aware Systems Department at the Institute for Infocomm Research, where he leads research in connected-home and context-aware systems. His research interests include pervasive computing, service-oriented computing, and context-aware systems. He received his PhD from the University of Rome La Sapienza and University of L'Aquila, Italy. Contact him at the Inst. for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613; daqing@i2r.a-star.edu.sg.



Xingshe Zhou is a professor in the School of Computer Science, Northwestern Polytechnical University, P.R. China. His research interests include distributed computing, embedded computing, and sensor networks. He received his MS in computer science from Northwestern Polytechnical University. He's a member of the IEEE. Contact him at the Northwestern Polytechnical Univ., MailBox #404, Xi'an, P.R. China, 710072; zhouxs@nwpu.edu.cn.

Chung-Yau Chin is a graduate student at the National University of Singapore. His research interests include pervasive computing, context-aware systems, service discovery, and the Semantic Web. He received his BE in computer engineering from the National University of Singapore. Contact him at 4 Amber Rd., Amber Tower #03-04, Singapore 439852; chungyau.chin@nus.edu.sg.

Xiaohang Wang is a graduate student at the National University of Singapore. His research interests include pervasive and context-aware computing and the Semantic Web. He received his BS in computer science from Huazhong University of Science and Technology, China. Contact him at the Inst. for Infocomm Research, MailBox #B097, 21 Heng Mui Keng Terrace, Singapore 119613; xwang@i2r.a-star.edu.sg.

Ji Men is a graduate student at the National University of Singapore. His research interests include pervasive computing and context-aware systems. He received his BS in computer science from University of Electronic Science and Technology of China. Contact him at the Inst. for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613; g0404995@nus.edu.sg.