Collaborative Personalised TV Programming

Jacob Sparre Andersen
Tofta Teld
Tåsingegade 36, 3. tv., 2100 København Ø, Danmark
sparre@sslug.dk

Abstract

We propose a set-top box design for giving the owner access to a number of TV channels adapted specifically to his/her interests. The channels are composed based on reviews distributed in RSS format [1, 2, 4]. The reviews can originate from other users of similar devices as well as from other sources. The device trains an artificial neural network [6] to prioritise the available programmes based on the available reviews and the feedback from the user. The proposal attempts to minimise the need for user interaction, as well as the need for centralised, critical services.

1. Introduction

The possibility of caching a large number of TV programmes from different sources so they are available to the individual viewers on demand makes it possible to introduce something which effectively can be seen as personal TV channels. This paper presents one way of doing this based on ideas from podcasting, blogs and Usenet. The intent with the system presented here is to allow the users to get on-demand TV adapted to their personal preferences.

The critical element in the system is the *preference engines*, which have to be able to learn to identify which preferences of other users (and publishers) correspond with the user's preferences. The set-top box should contain several preference engines, corresponding to different virtual TV channels for different subjects or belonging to different users of the device.

By allowing the reviews of TV programmes to be distributed through multiple channels – HTTP [5], SMTP [3] and NNTP [7] – the system does not depend on a big, centralised database to identify the relevant correlations between reviews and relevance for the specific user.

This paper does not cover the detailed workings of an on-demand cache. We limit ourselves to assume that the on-demand cache is able to add programmes to its storage, and that it makes its choices as to which programmes to have available based on its interaction with the preference engines. Whether the on-demand cache fetches the programmes over an Internet connection, through a DTV receiver or using some other technology is not the subject of this paper.

Although this paper suggests some specific implementations and algorithms, it important to notice that the critical idea behind the system is to exchange reviews of TV programmes in a standard format and to use these reviews to adapt the virtual TV channels to the preferences of the individual viewers. The exact choice of learning algorithms is not critical, and the choice of using an artificial neural network simply reflects what the author expects will be most efficient to use.

The system is based on the proven technology of *podcasting*. Podcasting is basically the announcement of downloadable content using RSS formatted messages. The user then selects some of the announced content and downloads it onto an iPod (thus the name of the technology) or a similar device. Later the user can enjoy the content, even while being without Internet connection.

The RSS formatted messages are typically distributed by HTTP from the website of the person making the announcement, but nothing prevents the use of other distribution mechanisms, such as NNTP (newsgroups) or SMTP (e-mail).

Although podcasting usually refers to audio content downloadable through the HTTP protocol, nothing prevents us from letting the announcements refer to video or static content. Similarly it can also just as well refer to live TV or radio broadcast over the air or the Internet or to locally stored content such as a CD or DVD collection.

2. Interaction with the user

It is unlikely that the system will become popular, if it requires more user interaction than old-fashioned TV watching. Interested users should still have access to personalise the system, since one can expect that more specific reviews also are likely to provide more useful information for other

users' preference engines.

The daily interaction with the system should be manageable with a device similar to an ordinary TV remote controller. Virtual channels should be as easy to select as ordinary, old-fashioned channels.

In addition to channel selection, pause, fast forward, rewind, and other standard features on a TV remote controller, the device shall allow the user to give feedback on the quality of the choices made by the preference engines. This could for example be implemented as a limited number of buttons, each corresponding to a preprogrammed standard review. The reviews can either be fixed for all channels (simplifying the labelling of the buttons) or depend on the selected channel (requiring dynamic labels).

Each review consists of a number of RSS messages and directions for the system about where to send them, as well as direct feedback to the preference engine about which rating the programme should get ("lower", "higher", "higher than the previous programme", etc.). The RSS messages should as a minimum include an URL identifying the programme being reviewed, and a short string representing the viewer's opinion of the programme (for example "local news from Cagliari", "not good" or "interesting"). The RSS messages can be posted to the viewer's personal RSS feed, or be sent to a mailing list or newsgroup intended for distribution of reviews in RSS format.

A review action may in addition to submitting a number of reviews include skipping the remainder of the programme (typically because the viewer decides it is bad or boring).

2.1. Configuring the system

The configuration options for the system should be to create and name a number of virtual channels, and to define a number of review actions. The users of the system are not supposed to change the configuration frequently, so it is sensible to use a different interface than the TV remote controller for this purpose. Since the system will need a network connection for other purposes, it is logical to allow the configuration to be done over a network connection. The system should have two configuration interfaces. One should be based on HTTP and HTML, while the other should be based on an open API. The HTTP and HTML based interface will be useful for configuration of the system from any web-enabled device. The open configuration will on the other hand allow for the creation of special purpose configuration tools.

The creation of a virtual channel includes a decision about where it should get its input from. These inputs can both be announcements of available programmes ("podcasts") and reviews of programmes distributed by other sources. For HTTP and NNTP based RSS feeds this can be done by "bookmarking" a number of URLs for each virtual channel. For SMTP based RSS feeds, it is slightly more complicated, since the set-top box will need to configure an e-mail address for each virtual channel and subscribe this address to the wanted RSS feed mailing lists.

2.2. Extended interface

The very simple TV remote controller interface for interacting with the system is useful for TV style viewing, but nothing really prevents us from using the same kind of preference engines in more demanding settings. On a device with text entering capabilities, it would be logical to extend the standard reviews with a longer text describing the programme. There is also the option of extending the number of handled formats to go beyond video. Text based news and audio podcasts can just as well be filtered using the same tools as those we propose for generating virtual TV channels.

3. Interaction with the on-demand cache

The preference engines should interact with the cache in three important ways:

- Keeping track of which programmes are currently available for on-demand viewing.
- Through requests to have specific programmes downloaded to the cache.
- Answering queries from the cache system about the likelihood that there is interest in watching a programme.

The reason for keeping track of which programmes currently are available is to avoid attempting to offer the user to watch a programme which isn't available (yet).

The requests to have specific programmes downloaded to the cache should be based on the evaluation of the incoming reviews. Once a preference engine finds out that the incoming reviews indicate that the user is likely to want to watch a programme, it should send a request to the on-demand cache to have it downloaded.

Since the on-demand cache may get temporary access to download some programmes (i.e. through normal terrestrial TV broadcast), there should also be a possibility of querying the preference engines about the likelihood that there is interest in watching a programme.

The likelihood that there is interest in watching a programme depends on the rating of the programme on a virtual channel (i.e. by a specific preference engine), the typical time spent watching that channel, and the typical rating threshold at which the channel is left. To maximise the

useful information available to the on-demand cache for deciding which programmes to store, these three numbers and their standard variations are reported for each virtual channel, when the system informs the on-demand cache about a programme. [1]

The on-demand cache can in practice be both a completely independent unit with a sufficiently high bandwidth connection to the set-top box, and another piece of software running in the same box as the preference engines.

4. Interaction with other similar devices

The interaction of a preference engine set-top box with other similar devices will basically happen through the exchange of reviews in RSS format.

On the output side of things this means that the device should be able to send e-mails in RSS format, post RSS formatted messages to newsgroups, and be able to update web-based RSS feeds. Whenever the user activates one of the feedback selections on the remote, a number of reviews corresponding to that selection should be sent to the designated RSS feeds.

On the input side the device should be able to maintain a number of mailing list subscriptions per preference engine, as well as querying web-based RSS feeds, and reading newsgroups. The SMTP communications may pose some complications in the configuration of the system, while the HTTP and NNTP based feeds are likely to be close to trivial since the feeds can be specified using URLs.

The complications related to automating the configuration of SMTP based distribution makes it tempting to omit this feature, but since the possibility of controlling who gets to see which of your reviews is likely to be of concern to many people, it should not be forgotten.

Since the system will ignore e-mail which isn't in RSS format, and since the training system automatically will ignore irrelevant reviews, regular spam is not likely to be a serious issue for the preference engine e-mail addresses.

5. Privacy issues

It is worthwhile to notice that it is unique that this system does not **require** viewers to share their preferences with others. The system still depends on **some** viewers sharing their reviews with other viewers. It may in practice be enough to have access to a few professional reviewers' comments to get the preference engines to work acceptably.

Once a basic version of the set-top box has been implemented and deployed, it will be possible to measure how many reviews are needed to reach acceptable success rates, and what the optimal balance between reviews and other kinds of input to the preference engine is. This will determine how many users are needed to be willing to share their reviews for the system to work.

6. Input for the preference engines

These are the types of input we will make available for the preference engines:

- **Reviews**: The presence/absence of a review with a specific key by a specific reviewer.
- **Programme features**: For which languages there are soundtracks/subtitles. The original language(s) of the programme. Running time of the programme.
- **Internal data**: If the user already has seen the programme. How long time since last time the user saw the programme. How large a fraction of the programme the user actually watched.
- **User feedback**: The user's response to the rating of the programme.

Although some people considers the categorisation of a programme (thriller, romance, ...) a feature of the programme, we consider it a kind of reviews of the programme, since there isn't an objective way to categorise a programme. For practical purposes will non-numeric features of a programme, such as the name of the director of a film, similarly be viewed as a kind of review of the programme.

The user feedback is not used as input to the ANN, but rather as corrections to the output, when the ANN is being trained. The user feedback will be stored as information about which programme should have a higher ranking than which programme. This ranking information will be time-stamped, so more recent decisions can take precedent over earlier decision. The system will include one special "programme" which represents the threshold between programmes worth watching and programmes not worth watching.

A preference engine will only be allowed a limited number of inputs. Once the limit is reached, the least influential input is removed and substituted with a random choice from the pool of available inputs.

The influence of each input calculated to a preference engine is calculated as an average over the stored set of training data for that preference engine. For each input vector,

[1] We suggest that the cache allocates space to the different channels based on how much time is spent watching them. The allocated space should be filled with programmes which have ratings above the threshold level (minus the standard deviation times some multiplier). The remaining space could for example be used for a random selection of programmes which the preference engines then could use to learn more about the preferences of their users.

the absolute value change in the output from the preference engine which results from switching each of the elements in the input vector off is calculated. Absent elements in the input vector will thus score zero for that input vector.

In case multiple inputs have the same, lowest influence (most likely zero), we should either remove all or a randomly picked one of them.

In addition to the ranking data, sets of available input data about reviewed programmes will also be stored for training purposes. To limit the stored volume, input data sets are only stored when the preference engine decides to show a programme and when the on-demand cache queries the preference engine about a programme.

It will probably be a good idea to include decaying connections as a part of the training algorithm. This generally gives the benefit of simplifying the ANN and clarifying which inputs are not important.

7. Timing considerations

Training ANNs can be a computationally heavy process. This means that the system will have to take some care about when and how it trains the preference engines.

If play-back of video does not use the full CPU resources of the set-top box, the training can simply be run on a low priority in the background. Otherwise the training will have to be postponed to periods when the set-top box is not being used. This can for example be before the systems powers down after play-back.

Since the training may be underway, when there is need for a decision by the preference engine, it is important that the training algorithm doesn't overwrite the active rules before the new rules are good enough to replace them.

If the user requests a power-down while training is underway, the system should ask to get to keep running until training is finished. As long as the set-top box doesn't need to have any moving parts (disk and cooling fan) active when it is training the ANNs, this shouldn't be a problem.

8. Economic impact

The design of preference engines and user interfaces for this kind of system is clearly an area, where there is space for several different providers. One can imagine that the competitive edge of a future TV is based on the quality of its preference engine. To force device manufacturers to compete on features like the quality of their preference engines, it is important to agree on common, open formats and protocols for exchanging and distributing reviews (like the broadcast TV signals nowadays also use the same open standard(s) for all TV channels).

Opening up for increased user choice in TV programming, will limit how intruding stand-alone commercials inserted in the content can get away with being. This means that advertisement funded TV producers will be forced to adapt their financing model for the content distributed through the presented system. The adaption may be anything from a slight adjustment of how annoying the viewers perceive the commercials to be, through pay-per-view, to a completely revolutionary financing model.

Pay-per-view delivery of content is likely to be impacted by the presented system, since it will introduce the possibility of the payment for the content being considered by the preference engine and not by the viewer. It is not certain in which direction this impact will tend.

9. Deployment

As the design of the system is presented here, it will only be of use in locations with Internet access. In addition, the initial configuration of the system will require using a web browser. This limits the target group, but more and more people get computers and Internet access at home.

Since the system depends on collaboration with other users, it will only have a chance of becoming really useful, once there is a sufficient number of active users. How large this number is, is not clear at this time. Even when there is only very few users, the system will still be able to perform as well as property-based filtering, so the quality of the filtering is not a point the system is likely to fail on.

The precise design of the user interface, including response-times, is critical for user acceptance. A remote controller with dynamic labels which automatically adapt to the standard reviews for the selected channel is one option, which should be studied despite the added cost and complexity.

Given the size of the TV industry, it is possible that the response to this system will be an attempt to make it unusable, instead of adapting to it. These attempts may include "spamming" the system, with fake reviews. Since the preference engine only cares about correlated reviews, this should not interfere with the steady state of the preference engines, but it is likely to slow down the learning rate. It may be necessary to allow users to limit the set of acceptable review sources, to counterbalance such an attack on the system.

10. Property-based filtering

One of the competitors to the system proposed here is property-based filtering. Since we include the properties of programmes as a special case of reviews, we might claim that the proposed system always will be at least as good as property-based filtering. In reality there are a few differences:

Property-based filtering is usually proposed as a distributor-side operation. This implies that the devices deployed to the users can be much simpler and cheaper. Centralised calculation of preferences also reduces the possibilities of "spamming" the preference calculation.

Property-based filtering systems appear to have a rather closed nature. This limits competition and thus also innovation. As the system proposed here has an open nature, it is much more likely to spur further innovation and improvement of preference based TV delivery.

11. Conclusion

Most of the technology for this project is already available and proven. Similarly there is already lots of content available for use through this technology. What we still lack to have the presented mode of collaborative personalised TV programming is:

- Implementation of a preference engine.
- A user interface for showing the selections of the preference engine and for feeding the user selections to the rest of the world and back to the preference engine.

Both of these tasks are manageable within a relatively limited budget, even if one decides to include the design of system specific, user-friendly hardware in the project.

Given the large impact personalised TV programming is likely to have on the satisfaction and knowledge people get out of the available TV content, and the limited resources needed to create a working version of the system, I expect that the project will be a success.

11.1. Acknowledgements

Thanks to Ilaria Sanna, Susanna Lenzu, Morten Sparre Andersen, Hanne Munkholm, Maurizio Agelli and Stig Rejnholt Andersen for fruitful discussions about the presented project.

References

[1] RSS 0.91 spec, revision 3. Technical report, 1999.
[2] RSS 1.0 specifications. Technical report, December 2000.
[3] Simple mail transfer protocol. Technical report, April 2001.
[4] RSS 2.0 specifications. Technical report, 2003.
[5] R. Fielding, J. Gettys, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext transfer protocol – HTTP/1.1. Technical report, June 1999.
[6] J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the theory of neural computation*. 1991.
[7] B. Kantor and P. Lapsley. Network news transfer protocol. Technical report, February 1986.