An Integrated Approach to TV & VOD Recommendations

tvgenius.net

Introduction

Finding compelling shows to watch is the lifeblood of any TV service: when viewers can find content they love, satisfaction and usage both rise.

However, the amount of TV content being produced is now far greater than our ability to process it - Choosing what channel or show to watch has become a daunting task. As a result, 90% of viewing occurs on just the top ten channels.

There are several key problems that a recommendations engine solves. From a consumer perspective, it helps users cut through the clutter and quickly find interesting TV. This increases service satisfaction, and makes consumers feel they are getting good value.

From a business perspective, recommendation engines reduce churn, and help TV operators effectively promote premium programming, and generate higher incremental revenues.

With recommendations technologies, viewers are automatically exposed to TV and Video on Demand (VoD) content that they might like, without the need to search for specific items or browse a TV guide.

This white paper reviews the different underlying technologies and algorithms that can be used to provide automatic recommendations for TV and VoD content. It then goes on to examine the benefits of combining these approaches into an integrated recommendations solution based on multiple sources of data in a transparent user experience.

Market drivers for recommendations

The reduced costs of television production and transmission combined with the ever increasing back catalogues of the major broadcasters had lead to an explosion in the amount of content available to viewers.

Many TV operators have used this is a point of differentiation, each claiming to have more content available to watch than any of their competitors.

Unfortunately, the viewer is frequently left frustrated by a bewildering array of choices. As the amount of content and channels has increased, TV guide technology has not been able to keep pace, and offer modern content discovery journeys.

Recommendations engines are increasingly becoming a popular choice for solving the problem of content discovery. Recommendation solutions enable the user to find personally relevant content that they might not have known was available.

A successful recommendation engine

A successful recommendation engine fulfils several purposes. At the most basic level, a successful recommendation means that the user's interest was peaked, and they went on to interact with the content.

From a user's perspective, recommendations effortlessly introduce them to new TV content that keeps them interested and engaged. Equally, recommendation engines benefit TV operators by allowing them to capitalise on the 'long tail' of their content libraries, allowing them to offer relevant content to each individual. This results in increased satisfaction, higher ARPU, and reduced churn.

The challenges for recommendations

The difficulty in implementing recommendations is that different users have different tastes and opinions about which television programmes they prefer.

- Quality – a substantial portion of the shows that are recommended to the user should be shows that they would like to watch, or at least might find interesting.
- Transparency – it should be clear to the user why they have been recommended certain shows so that if they have been recommended a show they don't like they can at least understand why.
- User feedback – people are fanatical about television programmes and if they are being recommended a show that they don't like they should have an immediate way to say that they don't like it and subsequently never have it recommended again.
- Driving take-up – ultimately the recommendations needs to drive the take up of the shows that they are recommending. This can only be measured by monitoring the shows that are recommended and seeing how user behaviours change.
- Explainability – users privacy should be protected, and it should be clear why they are recommended a specific TV show.
- Data sparcity – accurate recommendations need to be served, even when there are gaps in the data, or a user is new.

The different approaches to recommendations

Most recommendations systems use a combination of different approaches, but broadly speaking there are three different methods that can be used:

- Collaborative filtering of different users' behaviour, preferences, and ratings
- Automatic content analysis and extraction of common patterns
- Social recommendations based on personal choices from other people

Each of these approaches to TV recommendations has its limitations. As a result, many recommendation solutions take a hybrid approach, using information from each of these different methodologies to define what shows are recommended to users.

Collaborative filtering

Used since the mid-1990s. collaborative filtering is one of the most common approaches to building recommender systems.

Collaborative filter methods are based on collecting and analysing a large amount of information on users' behaviour, activity or preferences. For example, User A is matched against a database to discover 'neighbours', or which other users who have historically similar to taste to User A. Items that the neighbours like are then recommended to User A, since she will probably like them.

Figure 1 - Passive collaborative filtering from Amazon

The most famous collaborative-filtering algorithm is Amazon's approach which provides recommendations for future purchases based on the users' past activity.

Such an approach is called "passive filtering" because it provides recommendations based on activity without explicitly asking the users' permission to do so.

It is also possible to implement "active filtering". This is where the user provides information that will be used as the basis for recommendations – such as rating their favourite shows. Active filtering approaches are common amongst DVD rental services that use ratings of previous purchases as part of the recommendation algorithm. One of the most famous active collaborative-filtering systems is Netflix's recommendations engine which has been advertised through the Netflix prize. In 2008 Netflix offered a million dollar prize to any team who could improve the recommendation engine's quality based on a novel approach to collaborative filtering.

The main problem with active collaborative filtering for TV shows is that viewers will only rate a show after watching it. Due to the lean back nature of TV, there has

tvgenius.net

been limited success in getting users to build unique profiles that are needed for this type of recommendation algorithm.

Passive filtering is less problematic when collecting the data, but requires substantial processing in order to make the data attributable to a single user.

Viewing information is typically not usable, but information on which shows people have clicked on within the electronic programme guide (EPG) or any favourite shows that users have highlighted provide an excellent basis for passive filtering. The major disadvantage of passive filtering is that users cannot easily specify which information they want to have used for recommendations and which they don't. As such, any information used for passive filtering must be carefully governed by a set of business rules to reduce the potential for inappropriate recommendations.

The final element to consider in passive filtering is the degree to which it amounts to user profiling. Many European countries have a strong culture of data privacy and every attempt to introduce any level of user profiling can result in a negative customer response

Content analysis

Besides collaborative filtering, content-based filtering is another common class of recommender systems. Content-based recommender systems make recommendations by analysing the content of textual information and finding similarities in the content.

In the TV world, the only content-analysis technologies available to date rely on the metadata associated with the programmes. However, the recommendations are only as good as the metadata, which often lacks richness. As a result, content analysis recommendations are usually based on genre or a certain actor. For example, Johnny Depp stars in *Sweeney Todd* and *SpongeBob SquarePants*, so the two pieces of content would be associated.

Keyword extraction by term frequency analysis

TheTDF-IDF(term frequency–inverse document frequency) weight is often used in information retrieval and text mining and can be used to extract keywords from TV show metadata.

This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus, in this case the metadata of a particular show in the set of metadata for all shows.

The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.

The term count in the given document is simply the number of times a given term appears in that document. This count is usually normalized to prevent a bias towards longer documents (which may have a higher term count regardless of the actual importance of that term in the document) to give a measure of the importance of the term ti *within the particular document* dj*. Thus we have the term frequency, defined as follows.*

$$\mathrm{tf_{i,j}} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

where ni,j *is the number of occurrences of the considered term* ti *in document* dj *and the denominator is the sum of number of occurrences of all terms in document* dj*.*

The inverse document frequency is a measure of the general importance of the term (obtained by dividing the number of all documents by the number of documents containing the term, and then taking the logarithm of that quotient).

$$\mathrm{idf_i} = \log \frac{|D|}{|\{d : t_i \in d\}|}$$

where |D| *is the total number of documents in the corpus.*

$|\{d : t_i \in d\}|$ *is the number of documents where the term* ti *appears (that is $n_{i,j} \neq 0$). If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to use* $1 + |\{d : t_i \in d\}|$

Then $(\mathrm{tf\text{-}idf})_{\mathrm{i,j}} = \mathrm{tf_{i,j}} \times \mathrm{idf_i}$

A high weight in tf–idf *is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights hence tend to filter out common terms.*

The usual approach for extracting important information from content is either to use marked up metadata, with keywords such as actors' names and directors or to use a term-frequency algorithm to automatically extract key words.

The advantage of content-analysis-based recommendations is that they provide quick recommendations across a wide range of content without having to worry about piracy issues, or gathering user activity information.

The disadvantage is that TV metadata is generally not generated with automatic recommendations in mind and is frequently not suitable or provides irrelevant recommendations.

Social Recommendations

The idea that people like to watch TV shows that other people have suggested to them is the basis of social recommendations. This can involve friends, family, TV critics, newspapers, and social networks like Twitter and Facebook.

Social recommendations differ from other recommendations in that they are not typically relevant to the user who is receiving the recommendation, other than based on their relationship to the person who made the recommendation.

Social-networking technologies allow for a new level of sophistication whereby users can easily receive recommendations based on the shows that other people within their social network have ranked highly. This provides a highly personal level of recommendations than can be achieved using editorial recommendations.

social recommendations can be served a variety of ways, including through specific mobile apps, companion devices, Facebook apps, and social TV guides.

A number of social networks dedicated to providing music recommendations have emerged over the last few years, such as Pandora, and Last.fm which tracks all of their listening habits, and then applies a collaborative filtering algorithm to identify similar users and serve recommendations.

With the dominance of sites like Facebook and Twitter, there are more personal services emerging with friends recommending specific programmes to each other. Social recommendations have already become a growth area as social networking tools become more sophisticated.

The advantage of social recommendations is that because they have a high degree of personal relevance, they are typically well received. Unfortunately, this also means that the suggested TV shows tend to cluster around a few well known or cult-interest programmes.

Bayes Theorem: An Integrated Approach to TV & VoD Recommendations

In order to integrate the various approaches to recommendations, it is necessary to derive a probabilistic map based on the likelihood that a viewer might like a certain piece of TV content. The mathematical basis for this is Bayes' theorem.

The basis of Bayes' theorem is predicting the probability that one event will occur given another. In the case of TV show recommendations this is typically the probability that the user would like a show given that they like another.

A Bayes network is a set of such conditional probabilities mapped together. Figure 2 of a Bayesian network for the UK's top TV shows during March 2011. In this case, the length of the line between the shows indicates the probability that a user who likes one show will equally like another.

Here we can see that some shows, like Doctor Who and The Bill act as popular nodes that have a high overall probability that users will like them because they are linked to many other shows. For simplicity, we term this a Relevance Map:

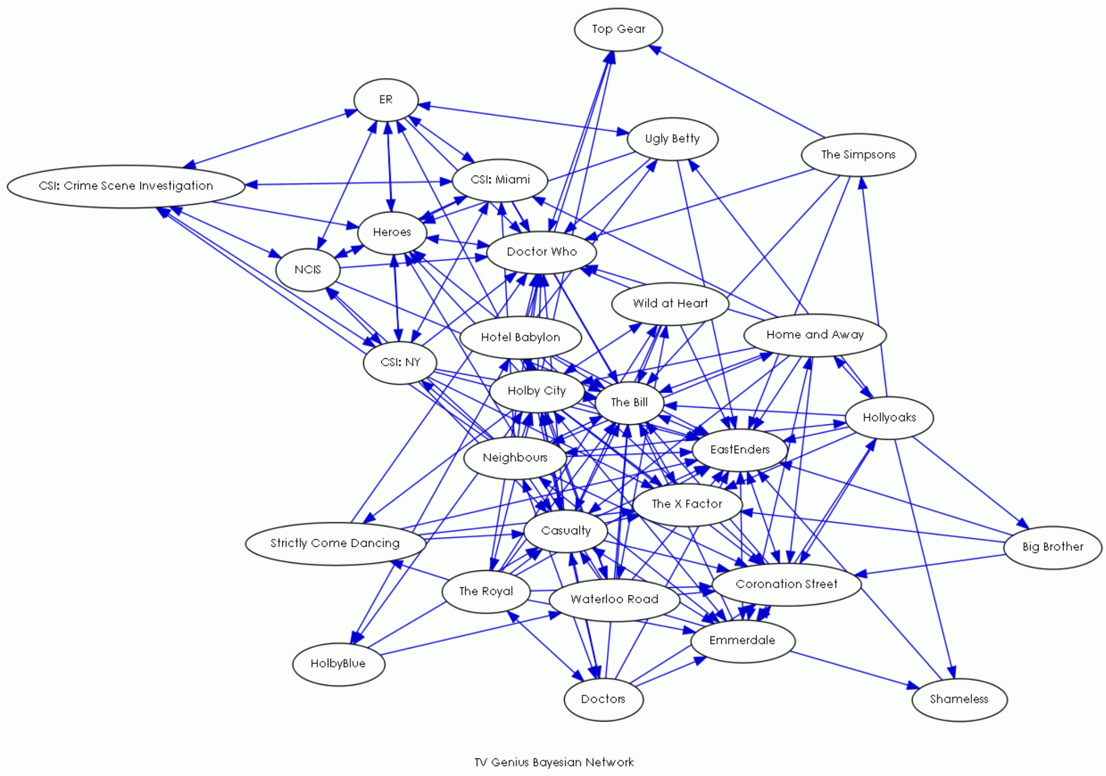

Figure 2 A Relevance Map showing popular shows on UK TV in August 2011

If we are to make recommendations based on a single show, we would highlight that show in the network and see which shows are adjacent to it, and then recommend these shows.

In Figure 3, the user likes the show Holby City, as shown below, then they can be recommended the show in grey, with Casualty, the Bill and Doctor Who being the most likely recommendations.

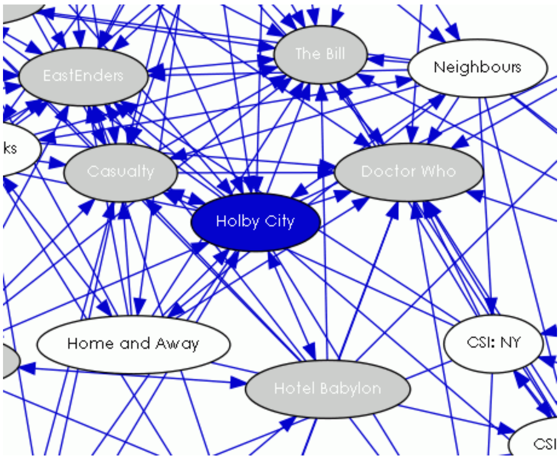

Figure 3 Recommendations for Holby City

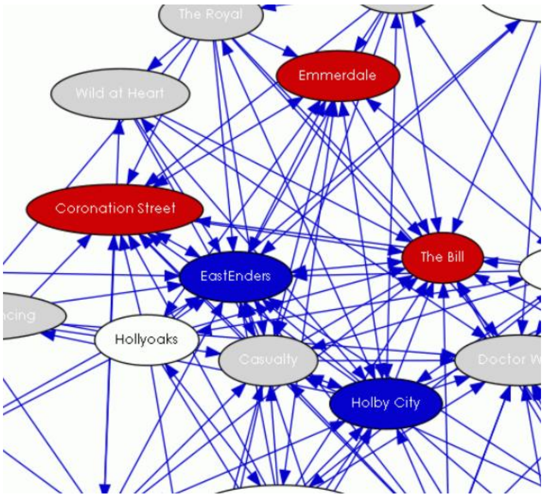

Figure 4 Recommendations for Holby City and Eastenders

If we further know that the user likes EastEnders, then the recommendations can be refined with the common shows between these two becoming the most likely show that the user would like, as shown in figure 4.

In this case, the most likely common shows between Holby City and Eastenders are Coronation Street and The Bill, and these would be the best shows to recommend.

Thus using Relevance Maps, a sophisticated level of recommendations can be provided on the basis of the TV shows that the user likes alone. This has the distinct advantage that there is no need to profile the user against the demographics, or activity – circumventing privacy concerns.

Indeed, using a Relevance Map, extremely accurate recommendations can be made from a simple list of favourite TV shows.

Building the Relevance Map

A Relevance Map can be produced through combining collaborative filtering, content analysis, and social recommendations, as well as incorporating other sources of information such as demographic information and viewing figures.

In building the network, these sources of information will need to be combined together with different weightings according the quality of the source of the information. Additionally, a half –life is typically assigned to each type of event.

TV Genius typically weighs any collaborative-filtering information based on favourite shows or active filtering as a strong driver in building the network with elements such as content analysis a weaker driver.

Once the network has been built it will normally need to be modified by business rules in order to prevent inappropriate recommendations and handle outliers. Inappropriate recommendations might typically be based around children's- and adult-specific content and are typically handled through the metadata of the TV shows.

Outliers are difficult to handle and are typically new shows that do not have sufficient history for a basis of content analysis and have no commonality on the basis of content analysis with existing shows. These would typically be handled on the basis of social recommendations but there are occasions where new shows are seeded into Relevance Map manually to prioritise them from day one.

Key benefits

The main difference between using a Relevance Map as the basis for recommendations and other approach is that all suggestions are made on the basis of television shows and in the context of TV service this has the following benefits:

- Transparency – there are no hidden variables behind how the recommendations are made. They are clearly associated with the shows that the user is looking at and/or has expressed an opinion about
- No barriers to entry – users do not need to enter any personal information or create a profile before receiving accurate recommendations. There is no need to monitor the user's activity or store a profile centrally
- Speed to market – any collaborative-filtering approach requires a large amount of information in order to build the network before recommendations can be sensibly made. Because the Relevance Map does not contain any user information it can be shared between businesses and used to provide recommendations from day one
- Easy to maintain – with no need for complex business rules or user segmentation, recommendations stay relevant and current without a large administrative overhead.

What to look for in a solution provider

There is a proliferation of companies offering recommendations services to the TV Industry at the moment. When looking for a platform it is important to consider the following:

Do they already have a solid Relevance Map to apply to your content?

Most providers are usually happy to share a snapshot of their Relevance Map under a non-disclosure agreement so you can get confidence that the service will be able to start making sensible recommendations from day one.

Is there a feedback mechanism between take-up and the network?

If some shows are particularly popular on your service, can this be fed back into the network to promote these shows further?

Are there existing application programming interfaces (APIs) that you can trial to test the service?

Trial the recommendations on a web or mobile service before you implement a full roll out.

Does it support promotion of individual content items?

As new shows launch there are always budgets available to support promotion and any recommendations engine should take this into account

What reporting is available?

You should expect if not real time, then at least daily reporting, showing the take-up of the recommendations, which are the shows that people are selecting the most and how this is driving take-up?

How much experience do they have?

It takes about two to three years of real-world deployments for most suppliers to tune their recommendations engines to a stage where they can be reliably deployed anywhere. Make sure that you are not the first implementation of a new recommendations system unless you are sure it is much better than other ones.

How transparent is the recommendations algorithm to the user?

Users don't like it when technology does things that they don't understand and are particularly sensitive to automatic profiling and targeted advertising. Make sure that the user experience makes it clear why the shows are being recommended to the user at all times.

How easy is it for the user to start using the service?

The more barriers there are to start using the service, the fewer people will use it. Avoid services that require the users to answer questionnaires, however short, before they can get going.

How much overhead will there be in managing the service from your side?

Do you want a recommendations engine that requires a team of back-end staff to manage the service or one that runs automatically based on user activity and feedback? Make sure you are clear on the staffing implications of the product you choose.

The TV Genius advantage

TV Genius is the premier provider of online TV discovery technology and the TV Genius Content Discovery Platform is used by millions of people across Europe to plan their TV viewing. Designed from the ground up as a platform to build TV guides, it has unprecedentedly rapid time to market and the flexibility to deploy and modify services as viewing habits change.

The geniusRecommend product is available on the TV Genius Content Discovery Platform and has the following advantages:

- Simple implementation – a full recommendations service can be deployed with very low levels of systems integration, reducing the need for internal resources to support a deployment and reducing time to market

- Fully managed service – there is no overhead in updating the product as TV services evolve and a full 24x7 service-level agreement ensures any problems are resolved in as little as 15 minutes

- Extensive client base – with a proven client base, TV Genius can be trusted to deliver. A sample of clients includes AOL, Arena, Bauer, BSkyB, BZ Berlin, Daily Mail, Freeview, IPC Magazines and ITV

- Low cost of ownership – with little systems-integration cost and a fully managed service, the cost of deployment is much less than comparable solutions that require extensive customisation and management

- Shared data set – with a large number of customers across Europe, TV Genius has built a Relevance Map based on a wide range of existing usage patterns that provide a high quality service without a long period for training the system

To learn more about how you could implement TV recommendations using geniusRecommend, please email sales@tvgenius.net