June 11, 2018

Refined music clustering

Christian Weitenberner

Ullas Gargi

Girum Ibssa

Brian Karlak

Krishmin Rai

Refined music clustering

ABSTRACT

Many different versions of the same song are often found in online music or video services. To enable easier search and better user experience, online music services often use clustering techniques to cluster different versions of the same song. Clustering systems are designed to be tolerant to shifts, noise, small tempo changes, etc. This tends to cause remixes and covers of a song to be clustered with the original versions of that song. This leads to certain problems, e.g., recommender systems selecting the most popular version of a song, in preference to other versions of the song; poor precision-recall rates of song metadata; etc. In turn, these lead to poorer experience for both users and creative partners of the music service.

This disclosure describes techniques that group music or videos into sub-clusters of increasing dissimilarity, based on signals that indicate provenance, quality, popularity, officiality, etc. The techniques enhance user experience by returning more appropriate search results.

KEYWORDS

- Online music
- Streaming music
- Audio streaming
- Recommender system
- Clustering
- Music clustering

BACKGROUND

Online music or video services often host many versions of the same song. Examples of this phenomenon are:

- popular songs that are recreated or remixed in the form of user-generated content (UGC), sometimes outnumbering the original by tens of thousands;
- cover versions of existing musical pieces;
- alternate audio-visual experiences of a song, e.g., a pure-audio version (typically created by a music-label partner of the online music service) and a music-video version (typically created as a promotional);
- live performance, studio recorded, official release, karaoke, etc. versions of a song; etc.

Clustering techniques are used to cluster different versions of a song to simplify song searches. However, such clustering of different versions leads to certain problems, such as:

- User-generated versions of a song, potentially of low quality, may out-rank the original, high-quality versions in popularity, leading to poorer recommendations;
- sub-par version selection, e.g., for a music-video, the pure-audio version may be played or cast to a screen being viewed, while the video version may be played while using headphones or as background music;
- poor precision-recall rates of song metadata (artist, album, song, format); etc.

Coarse clustering, as currently practiced, thus leads to a less than optimal experience for both end-users as well as creative partners of the music service.

DESCRIPTION

The techniques of this disclosure perform a fine sub-clustering of musical clusters such that differing versions of music within a single cluster remain identifiable to search engines, recommender systems, and end-users. Here, the word “music” is to be understood to generally include audio and video, such as songs, speech, music, video, music video, and other media.

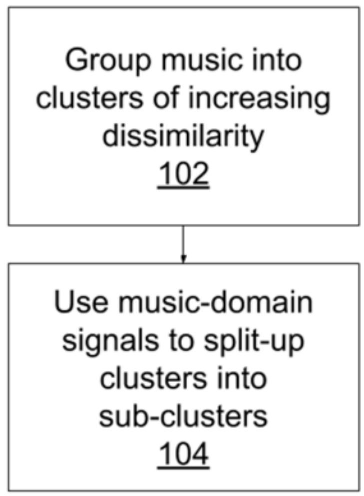

Fig. 1: Music clustering

Fig. 1 illustrates generation of refined sub-clusters of music. Music is grouped into concentric clusters of increasing dissimilarity (102). A given cluster includes different versions of a given music, e.g., pure audio, music-video, various recordings, substitutional (music-video with differing video), etc. The cluster may be formed using a threshold for similarity, e.g., an 80-80 similarity criterion indicates that any two musical pieces within a given cluster exhibit 80% similarity with each other.

Music-domain signals based on popularity, provenance, format, etc., are used to split up or fine-classify clusters into sub-clusters (104). For example, the fine-classification may be based on signals such as “live performance,” “official music video,” “official audio,” “remix version,” “version with lyrics,” “best cover,” “best live performance,” “karaoke,” “user-generated content," etc. Within each sub-cluster, music is ranked by popularity. Artist annotations are attached to each musical piece based on monetization claims or other signals. In this manner, different facets of the same music cluster are brought out, such that recommendation systems can find the best version of a given musical piece given user context.

Per the techniques of this disclosure, each sub-cluster is coherent in terms of artist annotation and type, e.g., each sub-cluster comprises similar music by a given artist. For ease of search, the highlights of each sub-cluster are generated and noted.

A given sub-cluster can include versions of music marked by potentially low quality, based, e.g., on signals such as UGC, claims, or uploader status. A sub-cluster may include versions of the music that are marked as official based on provenance, e.g., originating from a creative partner of the music service. When a recommendation system serves up a potentially low-quality version of a musical piece, it may advantageously be replaced by a higher quality version. In this manner, the techniques of this disclosure provide higher quality versions of a musical piece to a user. Further, the techniques conserve computing resources by avoiding automatic uploads of low-quality music from a user's local library if a high-quality match already exists. Still further, the techniques apply to infringement detection, e.g., by comparing the most popular UGC version of a song with what is recommended to a paying user of the service.

Each sub-cluster may include substitutional music-videos, e.g., videos that contain audio and/or video versions of the same song. For example, one version may include an audio track or a video containing just the audio recording with little or no imagery, or a static image. Another version may include rich video alongside the audio recording. A map is created between the two versions, thus enabling either version to be played based on user context. User context is determined upon user permission. For example, if it is determined that the user is currently using equipment with predominantly (or only) audio mode, e.g., headphones, then the audio-track version of the music-video is played. Conversely, if the user is currently using a video-enabled device, then a video version of the music-video is played.

The techniques enable the smearing (sharing) of metadata across music clusters or sub-clusters. Within a cluster, highlight music is chosen based on ranking by popularity or officiality signals. Metadata, e.g., artist, album, song details are transferred across similar musical pieces, e.g., those within a tight sub-cluster. In this manner, musical pieces without certain metadata benefit from metadata of other musical pieces.A higher reliability is assigned to videos with higher ranking.

Alternate to generating sub-clusters within clusters of music, as described herein, music-similarity measures with high precision may be used to cluster music that shows a very high degree of similarity. Such alternate techniques can be brittle, and do not leverage existing content fingerprinting techniques.

CONCLUSION

Many different versions of the same song are often found in online music or video services. To enable easier search and better user experience, online music services often use clustering techniques to cluster different versions of the same song. Clustering systems are designed to be tolerant to shifts, noise, small tempo changes, etc. This tends to cause remixes and covers of a song to be clustered with the original versions of that song. This leads to certain problems, e.g., recommender systems selecting the most popular version of a song, in preference to other versions of the song; poor precision-recall rates of song metadata; etc. In turn, these lead to poorer experience for both users and creative partners of the music service.

This disclosure describes techniques that group music or videos into sub-clusters of increasing dissimilarity, based on signals that indicate provenance, quality, popularity, officiality, etc. The techniques enhance user experience by returning more appropriate search results.