Big Data Visualisation and Visual Analytics for Music Data Mining

Katrina E. Barkwell[1], Alfredo Cuzzocrea[2], Carson K. Leung[1,*], Ashley A. Ocran[1], Jennifer M. Sanderson[1], James Ayrton Stewart[1], and Bryan H. Wodi[1]

[1] Department of Computer Science, University of Manitoba, Winnipeg, MB, Canada
[2] Department of Engineering and Architecture (DIA), University of Trieste, Trieste (TS), Italy
* E-mail: kleung@cs.umanitoba.ca

***Abstract*—As high volumes of a wide variety of valuable data of different veracities can be easily generated or collected at a high velocity nowadays, big data visualisation and visual analytics are in demand in various real-life applications. Musical data are examples of big data. Embedded in these big data are useful information and valuable knowledge. Many existing big data mining algorithms return useful information and valuable knowledge in textual or tabular forms. Knowing that "a picture is worth a thousand words", big data visualisation and visual analytics are also in demand. In this paper, we present a system for visualising and analysing big data. In particular, our system focuses on the big data science task of the discovery and exploration of frequent patterns (i.e., collections of items that frequently occurring together) from musical data. Evaluation results show the applicability of our system in big data visualisation and visual analytics for music data mining.**

Keywords—big data; data visualisation; visual analytics; visualiser; frequent patterns; musical data; music data analytics; music data mining

I. Introduction

Nowadays, high volumes of a wide variety (e.g., audio, video data) of valuable data of different veracities (e.g., precise data, imprecise and uncertain data) can be easily generated or collected at a high velocity in the current era of big data. The World Wide Web and music repository are excellent sources of big data. Embedded in these big data are rich sets of implicit, previously unknown, and potentially useful information and valuable knowledge, which can be discovered by big data mining [1, 2]. *Music data mining* [3-6], in particular, aims to discover implicit, previously unknown, and potentially useful information and valuable knowledge from musical data. It can be further categorized based on which of the following aspects of music to be mined:

- metadata (e.g., song title, band or singer's name, album name),
- genre (e.g., classical, folk, Jazz, Blues, country, hip-hop, rock, metal or heavy music),
- song or melody (e.g., pitch, rhythm), and/or
- lyrics (e.g., verses, choruses).

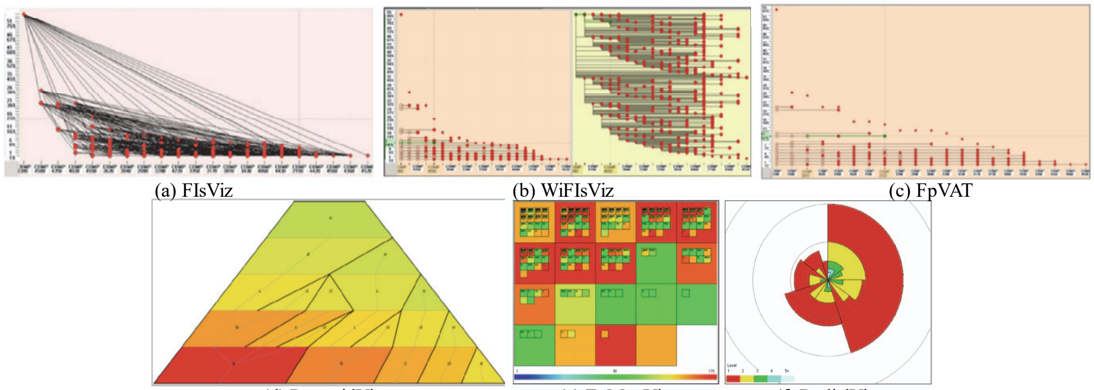
These, in turn, leads to research and development in areas like lyric text mining, cognitive musicology, computational musicology, and computational music analysis [7].

As one of the popular data mining tasks, frequent pattern mining finds frequently co-occurring items, events, or objects (e.g., frequently purchased merchandise items such as music albums in shopper market basket, frequently collocated music concerts). Applying frequent pattern mining on musical data finds frequent patterns. These mined knowledge and useful information reveals characteristics of the musical pieces, styles of songwriters (or music composers or lyricists), and preference of audiences or listeners. Over the past two decades, numerous frequent pattern mining algorithms [8, 9] have been designed and developed. Many of them have been focused on either functionality or efficiency. These algorithms usually return the mining results in textual form (e.g., a very long list of frequent patterns). Consequently, users may not easily comprehend the knowledge and useful information from the textual list.

When compared with textual representation of the data and the mined results, visual representation [10, 11] is more comprehensible to users. Some systems has been developed to visualise data (e.g., VisDB [12]) or missing data [13], as well as decision trees [14], association rules [15, 16], and clusters [17]). However, not too many visualisation tools have been developed to support frequent pattern mining. In the past few years, we [18-22] designed some tools and techniques to visualise patterns involving sets of items or related co-occurring entities. A common characteristic among the visualisers that were designed to support frequent pattern mining (e.g., FIsViz [23]) is that they display the mined frequent patterns in a traditional two-dimensional rectangular space. Consequently, the users who face the unfavourable orientation may have difficulty in comprehending the frequent patterns (e.g., important information such as frequency is not easy to read from those rotated graphs) shown on the graph. Moreover, many existing visualisers (e.g., FIsViz [23]) do not clearly indicate the subset-superset relationships between items (e.g., {*a, c*} is a subset of {*a, b, c*}). For those visualisers that try to reveal these relationships (e.g., multi-circular graph [24]), they are not clearly show due to massive overlapping of curves.

To improve the situations, we present in this paper a system for visualising and analysing big data. Our *key contribution* of this paper is our system that focuses on the big data science task of the discovery and exploration of frequent patterns (i.e., collections of items that frequently

978-1-5386-7202-0/18/$31.00 ©2018 IEEE
DOI 10.1109/iV.2018.00048



(a) FIsViz (b) WiFIsViz (c) FpVAT

(d) PyramidViz (e) FpMapViz (f) RadialViz

Figure 1. Some existing visualisers: (a) FIsViz [23], (b) WiFIsViz [27], (c) FpVAT [28], (d) PyramidViz [31], (e) FpMapViz [32], (f) RadialViz [33].

occurring together) from musical data. Our system uses an orientation-free, circular layout to visualise and analyse frequently co-occurring lyric texts. It also clearly reveals superset-subset relationships among the popular lyric texts.

The organisation of the reminder of this paper as follows. The next section presents background and related works. Section III describes our big data visualisation and visual analytics system for music data mining. Evaluation and conclusions are given in Sections IV and V, respectively.

II. Background and Related Works

Development of effective visualisation systems for data mining has been the subject of many studies. This line of research can be sub-classified into two general categories:

1. systems for visualising data (e.g., VisDB [12], Polaris [25]), and
2. systems for visualising the mining results (e.g., systems that visualise decision trees [14], association rules [15, 16], and clusters [17]).

In the past few years, some tools and techniques have recently been designed to visualise patterns involving sets of items or related co-occurring entities [18-22]. For example, Wong et al. [26] designed visualisation tools for visualising topic association rules and sequential patterns appearing in documents. Their visual tools are similar to parallel coordinates, in which keywords appear on the parallel coordinate axes in the y-direction and the sequential index (temporal or others) on the x-axis. Similarly, Yang [16] designed a system mainly to visualise association rules (but can also be used to visualise frequent patterns) in a two-dimensional space consisting of parallel vertical axes. In his system, all domain itemset are sorted according to their frequencies and evenly distributed along each vertical axis. A frequent pattern consisting of k items (i.e., a k-itemset) is then represented by a curve that extends from one vertical axis to another connecting k such axes. As the frequency of such a pattern is indicated by the thickness of the curve, it is not easy to compare the frequencies of patterns.

To make the comparisons of pattern frequencies easier, FIsViz [23] visualises frequent k-itemsets as polylines connecting k nodes in a two-dimensional space with (x, y)-coordinates, in which domain items are listed on the x-axis and frequency values are indicated by the y-axis. The x-locations of all nodes in the polyline indicate the domain items contained in a frequent pattern Z, and the y-location of the rightmost node of a polyline for Z clearly indicates the frequency of Z. Hence, prefix-extension relationships can be observed by traversing along the polylines. See Figure 1(a). As polylines in FIsViz can be bent and crossed over each other, it may not be easy to distinguish one polyline from another. To solve this problem, WiFIsViz [27] and FpVAT [28] were designed. As shown in Figure 1(b), WiFIsViz uses two half-screens to visualise frequent patterns. Both half-screens are wiring-type diagrams (i.e., orthogonal graphs), which represent frequent patterns as horizontal lines connecting k nodes in a two-dimensional space (where the x-axis lists all the domain items). The left half-screen provides the frequency information by using the y-location of the horizontal line to indicate the frequency of the frequent pattern. The right half-screen lists all frequent patterns in the form of a trie. FpVAT [28] also uses wiring-type diagrams to visualise frequent patterns. However, FpVAT shows all the frequent patterns and their frequencies on the same full-screen as shown in Figure 1(c).

The above three visualisers show *all* frequent patterns. When handling big data, the number of frequent patterns to be displayed can be huge due to pattern explosion. To improve this situation, CloseViz [29] extends WiFIsViz and FpVAT by visualising only *closed frequent patterns*, which greatly reduces the number of displayed patterns without losing any frequency information. A frequent pattern Z is *closed* if no superset of Z has the same frequency as Z.

Moreover, there are also situations in which users may be interested in differences between the results returned from two database instances. For easy comparisons between two big data sets, ContrastViz [30] extends WiFIsViz and FpVAT by visualising and analysing all the frequent patterns, their frequencies, as well as *changes* in frequencies between the two (spatiotemporal) data sets.

Instead of polylines or wiring-type diagrams (i.e., orthogonal graphs), PyramidViz [31], FpMapViz [32], RadialViz [33] and its variant [34] use alternative design with emphasis on showing the prefix-extension relationships among the frequent patterns. Specifically, PyramidViz [31] visualises frequent patterns in a building block layout. As shown in Figure 1(d), the frequent 1-itemsets are located at the bottom of the pyramid, whereas frequent patterns of higher cardinalities are located near the top of the pyramid. Frequent patterns are represented in a hierarchical fashion so that the building blocks representing the extensions of a frequent pattern Z are put on top of the blocks representing the prefixes of Z. The colour of the block representing Z indicates the frequency range of Z.

Nowadays, in a collaborative environment, it is not uncommon for collaborators to have face-and-face meetings. Partially due to the emerging of table-top displays as an effective platform for collaboration, information is shared on the table-top surface in the meetings. As such, orientation or view perspective cannot be neglected. Unlike a single-user environment (where orientation may not be an issue), object orientation becomes critical in a multi-user environment because not all users share a common perspective of the displayed information. As information is viewed from different positions, it may be perceived differently. A study [35] showed that user perception (e.g., legibility or readability) of a chart decreases when the chart is not oriented right-side up. However, the user perception can be improved by using orientation-free visualisers. For instance, FpMapViz [32] represents frequent patterns as squares in a hierarchical fashion so that extensions of a frequent pattern Z are embedded within squares representing the prefixes of Z. The colour of the square representing Z indicates the frequency range of Z, and the size of the square indicates the cardinality of Z. See Figure 1(e). Similarly, as shown in Figure 1(f), RadialViz [33] also represents frequent patterns in a hierarchical fashion. However, a radial layout is used so that extensions of a frequent pattern Z are embedded within sectors representing the prefixes of Z. The frequency of Z is represented by the radius of the sector representing Z, the cardinality of Z is represented by the colour of the sector.

III. Our System for Big Data Visualisation and Visual Analytics

In this section, we present our system for big data visualisation and visual analytics. In particular, we focus on a music data mining task of *lyric text mining*. By applying frequent pattern mining algorithm to musical data, we can find collections of frequently co-occurring words or phrases in the verses or choruses. The mined results—in the form of frequent patterns—help reveal the following:

- characteristics of the musical pieces (e.g., what are some common words or phrases used in certain music genre?);
- styles of the corresponding songwriters, music composers, or lyricists (e.g., what are some popular words or phrases frequently put by the lyricists in their songs?); and
- preference of audiences or listeners (e.g., what are some words or phrases frequently liked by listeners of popular music?).

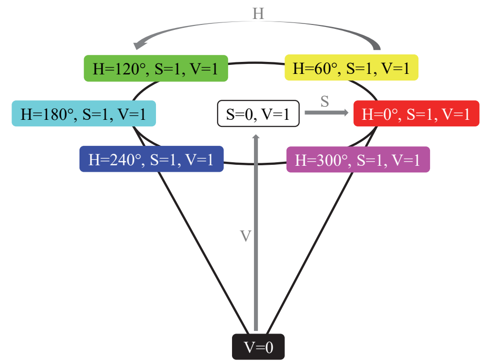


Figure 2. HSV colour model.

Once these frequent patterns are found, our **hue-saturation-value based visualiser**—called **HSVis**—shows these mining results. Specifically, HSVis uses a circular layout (which is orientation free) to directly show frequent patterns and their relationships (both prefix-extension relationships, as well as subset-superset relationships).

To visualise the hierarchical structure of frequent patterns (and their relationships) in a radial layout can be challenging because:

- Not all extensions of a frequent pattern Z are disjoint. In fact, extensions of Z are usually overlapping (e.g., as two extensions of $\{a, b\}$, both $\{a, b, c\}$ and $\{a, b, d\}$ are overlapping, where *a, b, c* & *d* are words or phrases in a song).
- The frequency of a frequent pattern Z is *not* necessary higher than or equal to the *frequency sum* of *all* extensions of Z.
- Fortunately, the frequency of a frequent pattern Z is still higher than or equal to the frequency of *each* extension of Z.

Based on these three challenges and observations, HSVis represents the frequent patterns as follows. To have a more systemic way to determine the colour (which represents a specific cardinality of itemsets), HSVis uses hue. In the hue-saturation-value (HSV) colour model, *hue* is a colour appearance attribute of a visual sensation. As hue ranges from 0° to 360°, our HSVis divides this range according to the number of different cardinalities. For instance, with six different cardinality, HSVis divides this 360°-range into six regions as shown in Figure 2:

1. the 0°-region that represents 1-itemsets (e.g., frequently occurring individual words or phrases in the lyrics) in red;
2. the 60°-region that represents 2-itemsets (e.g., frequently occurring pairs of words or phrases in the lyrics) in yellow;
3. the 120°-region that represents 3-itemsets (e.g., frequently occurring triplets of words or phrases in the lyrics) in green;
4. the 180°-region that represents 4-itemsets (e.g., frequently occurring quadruplets of words or phrases in the lyrics) in cyan;

Figure 3. Colour changes from a washed-out colour to a pure colour when *saturation* increases.

Figure 4. Colour changes from a pure colour to a dark colour when *value* decreases.

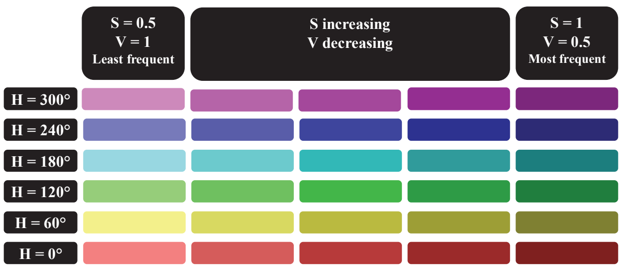


Figure 5. Colour changes for least frequent patterns (with S=0.5 & V=1) to most frequent patterns (with S=1 & V=0.5) for six cardinalities (*hues*).

5. the 240°-region that represents 5-itemsets (e.g., frequently occurring quintuplets of words or phrases in the lyrics) in blue; and
6. the 300°-region that represents 6-itemsets (e.g., frequently occurring quintuplets of words or phrases in the lyrics) in magenta.

To avoid any outlying objects or skewed data generating a spike, HSVis does not use the radius or distance from (or to) the centre to represent frequencies of itemsets. Instead, HSVis uses a combination of *saturation* and *value*. In the HSV colour model,

- *saturation* shows the colourfulness of a stimulus relative to its own brightness. In other words, saturation—expressed as a number between 0 and 1 inclusive—can be considered as the purity of a colour. The higher the saturation, the purer is the colour. Conversely, the lower the saturation, the more washed out (e.g., the whiter) is the colour. See Figure 3.
- *value* shows the brightness relative to the brightness (or darkness) of a similarly illuminated white. Value is also expressed as a number between 0 and 1 inclusive. The lower the value, the darker (i.e., blacker) is the colour. Conversely, the higher the value, the purer is the colour. See Figure 4.

When both saturation and value decrease, they become harder to distinguish for a specific hue. So, HSVis does not let the *saturation* or *value* fall below 0.5. When *saturation* = *value*, the colour tends to look very grey. Hence, HSVis choses to have a colour scale in which these two numbers varied from one another. The scale ranges from the least frequent items having *saturation* = 0.5 and *value* = 1. As the itemsets grow in frequency, their corresponding sections in the chart increase to *saturation* = 1 and decrease to *value* = 0.5. As such, the sum of these two numbers is always 1.5:

$$\textit{saturation} + \textit{value} = 1.5, \tag{1}$$

where *saturation* $\in$ [0.5, 1] and *value* $\in$ [0.5, 1]. This means that, on the one hand, the lowest-frequency itemsets would appear very washed out, but not at all dark. On the other hand, the highest-frequency itemsets would appear very saturated and relatively dark. As the sum ranges from (*saturation*=0.5 & *value*=1) to (*saturation*=1 & *value*=0.5), HSVis uniformly distributed among the frequencies in the frequent patterns mined from musical data. As shown in Figure 5, HSVis shows the most frequent 1-itemset in dark red, the most frequent 2-itemset in dark yellow, the least frequent 5-itemset in washed-out blue, and the least frequent 6-itemset in washed-out magenta.

To avoid stacking a superset over its subset, HSVis represents itemsets of different cardinality in rings. As we expected more frequent itemsets at lower cardinality than frequent itemsets at higher cardinality, HSVis "radiates in" these rings from 1-itemsets on the outermost (red) ring to the 6-itemsets on the innermost (magenta) ring. By so doing, if the frequency of the superset is the same as that of the subset, then the sector of the superset is no longer covered by that of the subset, and thus becomes visible.

Moreover, HSVis also provides users details-on-demand by allowing them to hover the mouse over an arc area to (i) highlight the arc area and (ii) display a small box showing the frequency of the corresponding frequent pattern represented by the arc area.

To facilitate easy lookup of frequent patterns, the default ordering used in HSVis is to arrange items anticlockwise in some user-specified order (e.g., alphabetical order). With such an arrangement, users can easily locate the patterns of interest. Furthermore, with this item arrangement, users can still easily spot the patterns with the highest or lowest frequencies. The reason is that HSVis uses combination of *saturation* and *value* to show frequency. A pattern with the high frequency is represented by the dark sector, whereas a pattern with the low frequency is represented by the washed-out (i.e., whiter) sector. Moreover, while the default order is the user-specified frequency independent ordering, HSVis also provides users with an option to display patterns in a frequency-dependent (e.g., decreasing frequency order). Frequent patterns of the same cardinality are represented by the same colour. Singleton are represented in red, pairs are in yellow, triplets are in green, quadruplet are in cyan, quintuplets are in blue, and sextuplets are in magenta. In the design of HSVis, each itemset is given a designated sector of the entire circle, but the shape representation of each level of cardinality differs significantly. For each itemset, the high-cardinality levels are represented by smaller sectors of smaller circles. The area is a section of donut shape or ring, which leaves its inner circles open for the next levels of cardinality. This means that itemsets of cardinality 1 are represented by the outermost ring of the circle. The next highest level of cardinality, the itemsets of cardinality 2 are represented by a portion of the second outermost ring, and so on moving inward. Itemsets of each cardinality will be represented in the same sector of the circle as their supersets, whose items will be the prefix of their itemsets. Since the location of each higher cardinality level is dependent on its prefix, {*a*} and {*b*} may be frequent, but if {*a, b*} is also frequent, then the section representing AB would be shown in the sector of {*a*} and not of {*b*}, as to avoid duplication.

When designing this visualiser, it became apparent that itemsets of high cardinality would be difficult to distinguish at their innermost levels. Hence, a weight is given to each sector based on the cardinality level of each subsequent superset having that itemsets as its prefix. At each cardinality level, each section is given weight of 1 to begin. Then, it is given an incremental weight of 1 for each cardinality level coming after it, meaning that there is more space left to be able to distinguish the innermost levels of cardinality toward the centre of the circle. Let us consider an example. Suppose that {*a*}, {*b*} and {*c*} are the only frequent singleton itemsets (i.e., common words in the lyrics), {*a, b*} is the only 2-itemset, and there are no more frequent itemsets. Then, {*a*}, {*b*} and {*c*} would each start with a weight of 1. Because {*a*} had a superset of one level higher (i.e., a superset of cardinality 2), its inner section (e.g., {*a, b*}) would be given an additional weight of 1 (i.e., a weight of 2). In other words, weights of {*a*} is 2, of {*b*} is 1, and of {*c*} is also 1. Since there is a total weight of 4 at this level of cardinality and {*a*} has twice the weight of the others, {*a*} will also have twice the *angular space* as the others. Half of the available space would be allocated {*a*}, and a quarter each would be allocated to {*b*} and {*c*} each. The available space will never exceed that of its subsets so that each itemset will always have its own sector of the original circle.

When HSVis shows all frequent patterns, it gives users an overview about the distribution of all frequent patterns. As some arc areas are small, HSVis provides users with interactive features to zoom in and zoom out so that users can obtain information at the granularity level of their interest. So far, we have described (i) how HSVis represents frequent patterns and (ii) how the layout of frequent patterns reveals the prefix-extension relationships—e.g., arc area for {*a, b, c, d*} is connected to that for {*a, b, c*}, which in turn is contained in that for {*a, b*} and for {*a*}. However, arc area for {*b*}—which is a *subset* but a *prefix* of {*a, b*}—is not connected to that for {*a, b*}. So, a logical question is how to reveal the subset-superset relationships? To answer this question, our HSVis provides users with interactive features to drill in some specific area of interest. By doing so, HSVis redraws the diagram and include the supersets. For instance, when interactively specifying the user interest in {*b*}, HSVis shows {*b*}, as well as {*a, b*}.

IV. Evaluation

We evaluated our HSVis by using the World Wide Web and music repository about top hit songs over the past decade, from which frequent patterns (e.g., collection of commonly used co-occurring words and phrases in popular musical data) were discovered. Then, our HSVis visualises and analyses these mined results. For example, when viewing HSVis as shown in Figure 6, we can find the following from musical data:

- Frequently occurring singleton words or phrases (e.g., {*you*}, {*it*}, {*my*}, {*be*}, {*with*});

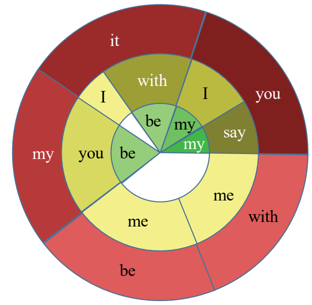


Figure 6. Our HSVis visualising frequent lyric texts.

- Frequently occurring pairs of words or phrases (e.g., {*you, say*}, {*you, I*}, {*it, with*}, {*it, I*}, {*my, you*}, {*be, me*}, {*with, me*}); and
- Frequently occurring triplets of words or phrases (e.g., {*you, say, my*}, {*you, I, my*}, {*it, with, be*}, {*my, you, be*}).

Our HSVis reveals that identifying words such as "you", "I" and "me" are the frequently occurring in lyrics. This lets data scientists or analysts know that the songs that are most popular involve content about people and their experiences.

Since words such as "you", "I" and "me" were universally common amongst all songs, we filtered them out so as to discover more descriptive/unique words. Consequently, HSVis shows the top sequences that make up popular music found by the user moving in toward the centre once the base word is selected.

In addition to applying lyric text mining to *all* songs, we also apply lyric text mining to different genres and compared the results among different genres. For instance, we took 10 top songs in the genres of country, rock, as well as metal or heavy music. We compared the diction that appeared frequently and found the following:

- Frequently occurring words in top country songs include "Jolene",
- frequently occurring words in top heavy metal songs include "dark" and "fear"; and
- frequently occurring words in top rock songs include "black".

V. Conclusions

In this paper, we presented our system for visualising and analysing big data for music data mining—especially, lyric text mining. Our hue-saturation-value (HSV) based data visualisation and visual analytics system called HSVis provides users with interactive circular representation and exploitation of frequent patterns. HSVis represents collections of mined frequently patterns using a radiating-in layout (which is orientation free) and in a hierarchical fashion (so that extensions of a collection *Z* are connected to the sector representing *Z*). Patterns of the same cardinality have the same *hue* (i.e., same colour), and patterns of different cardinalities have different *hue* (i.e., different colours). Since HSVis uses a balanced combination of *saturation* and *value* in the well-known HSV model to indicate the frequencies of patterns, users can easily observe the relative frequency distribution of all the patterns. Patterns having similar combined numerical sum of *saturation-value* have similar frequencies. With interactive features (e.g., mouse hover, zoom-in, drill-in), users can easily explore patterns of interest.

Although HSVis was designed for visualising and analysing musical data, it can be generalised to many other applications such as visual analytics of patterns like weighted frequent subgraphs [36] and popular "following" patterns [37], as well as visualisation of online game data [38].

ACKNOWLEDGMENT

This project is partially supported by Natural Sciences and Engineering Research Council of Canada (NSERC) and University of Manitoba.

REFERENCES

[1] C. K. Leung, "Big data analysis and mining," Encyclopedia of Information Science and Technology, 4e, 2018, pp. 338-348.

[2] C. K. Leung, F. Jiang, T. W. Poon, & P. Crevier, "Big data analytics of social network data: who cares most about you on Facebook?" Highlighting the Importance of Big Data Management and Analysis for Various Applications, 2018, pp. 1-15.

[3] I. Karydis, A. Gkiokas, V. Katsouros, & L. S. Iliadis, "Musical track popularity mining dataset: extension & experimentation," Neurocomputing 280(6), Mar. 2018, pp. 76-85.

[4] T. Li, M. Ogihara, & G. Tzanetakis, "Guest editorial: special section on music data mining," IEEE TMM 16(5), Aug. 2014, pp. 1185-1187.

[5] B. Martínez & V. Liern, "A fuzzy-clustering based approach for measuring similarity between melodies," in Proc. MCM 2017, pp. 279-290.

[6] K. Neubarth &, D. Conklin, "Contrast pattern mining in folk music analysis," Computational Music Analysis, 2016, pp. 393-424.

[7] D. Meredith (ed.), Computational Music Analysis, Springer, 2016.

[8] C. K. Leung, "Frequent itemset mining with constraints," Encyclopedia of Database Systems, 2e, 2016.

[9] L. Rompré, I. Biskri, & J. Meunier, "Using association rules mining for retrieving genre-specific music files," in Proc. FLAIRS 2017, pp. 706-711.

[10] E. Ventocilla & M. Riveiro, "Visual analytics solutions as 'off-the-shelf' libraries," in Proc. IV 2017, pp. 281-287.

[11] Y. Tanaka, J. Sjöbergh, & K. Takahashi, "A need for exploratory visual analytics in big data research and for open science," in Proc. IV 2016, pp. 261-270.

[12] D.A. Keim & H.-P. Kriegel, "Visualisation techniques for mining large databases: a comparison," IEEE TKDE 8(6), Dec. 1996, pp. 923– 938.

[13] J. Sjöbergh & Y. Tanaka, "Visualizing missing values," in Proc. IV 2017, pp. 242-249.

[14] M. Ankerst, C. Elsen, M. Ester, & H. Kriegel, "Visual classification: an interactive approach to decision tree construction," in Proc. ACM KDD 1999, pp. 392–396.

[15] J. Blanchard, F. Guillet, & H. Briand, "Interactive visual exploration of association rules with rule-focusing methodology, KAIS 13(1), 2007, pp. 43–75.

[16] L. Yang, "Pruning and visualising generalized association rules in parallel coordinates," IEEE TKDE, 17(1), Jan. 2005, pp. 60–70.

[17] M. R. Hassan, K. Ramamohanarao, C. K. Karmakar, M. M. Hossain, & J. Bailey, "A novel scalable multi-class ROC for effective visualisation and computation," in Proc. PAKDD 2010, pp. 107–120.

[18] P. Braun, A. Cuzzocrea, T. D. Keding, C. K. Leung, A. G. M. Pazdor, & D. Sayson, "Game data mining: clustering and visualization of online game data in cyber-physical worlds," Procedia Computer Science 112, pp. 2259-2268.

[19] P. M. J. Dubois, Z. Han, F. Jiang, & C. K. Leung, "An interactive circular visual analytic tool for visualization of web data," in Proc. IEEE/WIC/ACM WI 2016, pp. 709-712.

[20] C. K. Leung, "Data and visual analytics for emerging databases," in Proc. EDB 2017, pp. 203-213.

[21] C. K. Leung, C. L. Carmichael, Y. Hayduk, F. Jiang, V. V. Kononov, & A. G. M. Pazdor, "Data mining meets HCI: data and visual analytics of frequent patterns," in Proc. ECML-PKDD 2016, Part III, pp. 289-293.

[22] C. K. Leung, C. L. Carmichael, P. Johnstone, R. R. Xing, & D. S. H. Yuen, "Interactive visual analytics of big data," Ontologies and Big Data Considerations for Effective Intelligence, Feb. 2017, pp. 1-26.

[23] C. K. Leung, P. P. Irani, & C. L. Carmichael, "FIsViz: a frequent itemset visualiser," in Proc. PAKDD 2008, pp. 644–652.

[24] G. Bothorel, M. Serrurier, & C. Hurter, "Visualisation of frequent itemsets with nested circular layout and bundling algorithm," in Proc. ISVC 2014, pp. 396–405.

[25] C. Stolte, D. Tang, & P. Hanrahan, "Query, analysis, and visualisation of hierarchically structured data using Polaris," in Proc. ACM KDD 2002, pp. 112–122.

[26] P. C. Wong, W. Cowley, H. Foote, E. Jurrus, & J. Thomas, "Visualising sequential patterns for text mining," in Proc. IEEE InfoVis 2000, pp. 105–111.

[27] C. K. Leung, P. P. Irani, & C. L. Carmichael, "WiFIsViz: effective visualisation of frequent itemsets," in Proc. IEEE ICDM 2008, pp. 875–880.

[28] C. K. Leung, & C. L. Carmichael, "FpVAT: a visual analytic tool for supporting frequent pattern mining," ACM SIGKDD Explorations 11(2), Dec. 2009, pp. 39–48.

[29] C. L. Carmichael & C. K. Leung, "CloseViz: visualizing useful patterns," in Proc. ACM SIGKDD 2010 Workshop on UP, pp. 17–26.

[30] C. L. Carmichael, Y. Hayduk, & C. K. Leung, "Visually contrast two collections of frequent patterns," in Proc. IEEE ICDM 2011 Workshops, pp. 1128–1135.

[31] C. K. Leung, V. V. Kononov, & A. G. M. Pazdor, "PyramidViz: visual analytics and big data visualization of frequent patterns," in Proc. IEEE DASC/DataCom 2016, pp. 913-916.

[32] C. K. Leung, F. Jiang, & P. P. Irani, "FpMapViz: a space-filling visualization for frequent patterns," in Proc. IEEE ICDM 2011 Workshops, pp. 804-811.

[33] C. K. Leung & F. Jiang, "RadialViz: an orientation-free frequent pattern visualizer," in Proc. PAKDD 2012, Part II, pp. 322–334.

[34] P. M. J. Dubois, Z. Han, F. Jiang, & C. K. Leung, "An interactive circular visual analytic tool for visualization of web data," in Proc. IEEE/WIC/ACM WI 2016, pp. 709-712.

[35] F. Alallah, D. Jin, & P. Irani, "OA-graphs: orientation agnostic graphs for improving the legibility of charts on horizontal displays," in Proc. ITS 2010, pp. 211–220.

[36] M. A. Islam, C. F. Ahmed, C. K. Leung, C. S. H. Hoi, "WFSM-MaxPWS: an efficient approach for mining weighted frequent subgraphs from edge-weighted graph databases," in Proc. PAKDD 2018. doi: 10.1007/978-3-319-93040-4_52

[37] C. K. Leung & F. Jiang, "Efficient mining of 'following' patterns from very big but sparse social networks," in Proc. IEEE/ACM ASONAM 2017, pp. 1025-1032.

[38] P. Braun, A. Cuzzocrea, T. D. Keding, C. K. Leung, A. G. M. Pazdor, & D. Sayson, "Game data mining: clustering and visualization of online game data in cyber-physical worlds," Procedia Computer Science 112, 2017, pp. 2259-2268.