Visualizing hidden structures in datasets using deep learning

Dmytro Perekrestenko

Supervisor: Xavier Bresson

Professor: Pierre Vandergheynst

June 1, 2015

1 Introduction

The goal of this project is to find hidden structures in large-scale and high-dimensional datasets. Deep Learning has shown great promise to extract meaningful features for data classification. In this project we leverage this feature identification technique to visualize patterns of interest that are usually hard to access. Specifically, we applying non-linear dimensionality reduction techniques (such as t-SNE) to the deep learning features and reveal essential patterns characterizing the dataset. As an application, the project focus on the well-known images benchmark MNIST dataset, well-known audio benchmark GTZAN dataset and Montreux Jazz Festival archives. The deliverable of the project is matlab code that solves 3 tasks - visualisation of ambiguous datasets, structure mining and music recommendation.

2 Project Goal

The goal of the project is to develop a tool to find hidden structures in large-scale and high-dimensional datasets. The idea of the project is briefly described in the figure 1.

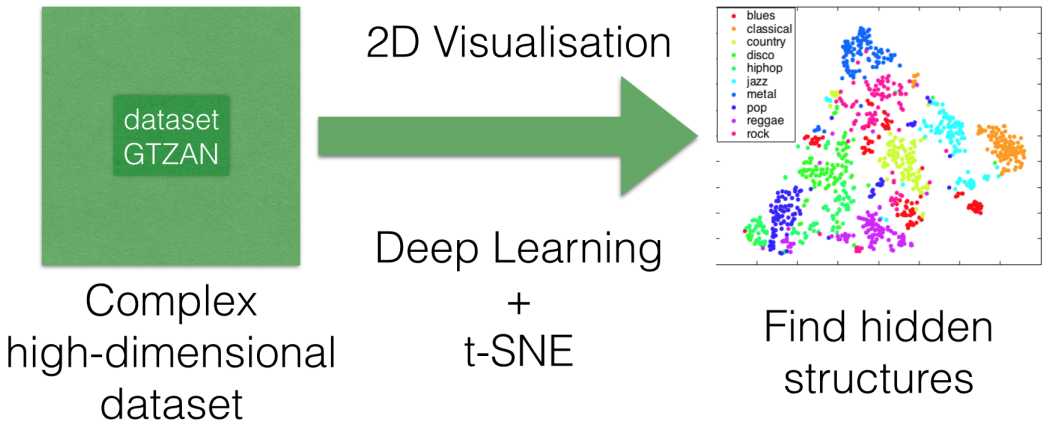

Figure 1: Project structure

The system is given high-dimensional complex dataset (e.g. GTZAN), which we process with combination of Deep Learning and t-distributed Stochastic neighbour embedding algorithms to get a 2D projection with visible structure of the data. Then we analyze obtained "good" 2D projection to mine the structural properties of the data.

3 Features

Since the project is concentrated mainly on music datasets and raw audio data is too redundant for visualisation algorithms to process directly, we first extracted features from it. We used two types of features - handcrafted and learned. As handcrafted features we used features described in (Sturm, 2013) and Temporal Echonest features. As learned features we used ones obtained by dictionary learning on audio spectrograms.

3.1 Handcrafted (Sturm '13)

In paper (Sturm, 2013) there are two types of features - short term and long term. Short term features are computed on small segments in time. These segments are called analysis windows and have to be small enough so that the frequency characteristics of the magnitude spectrum are relatively stable. However, the sensation of a sound "texture" arises as the result of multiple short-time spectrums with different characteristics following some pattern in time. For example, speech contains vowel and consonant sections which have very different spectral characteristics. Therefore, in order to capture the long term nature of sound "texture" long-term features computed on larger "texture" windows are used.

Short-term audio features

1. Octave-based spectral contrast (OSC) - was developed to represent the spectral characteristics of a music piece. It considers the spectral peak and valley in each sub-band separately. In general, spectral peaks correspond to harmonic components and spectral valleys correspond to non-harmonic components or noise

in a music piece. Therefore, the difference between spectral peaks and spectral valleys will reflect the spectral contrast distribution.

2. Mel-frequency cepstral coefficients (MFCCs) - are coefficients that collectively make up short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency.

3,4. Spectral centroid and spectral rolloff
Spectrogram $X(\omega;t)$ is normalised such that it sums to one, to produce the probability mass function $p_X(\omega;t)$ and cumulative distribution function $P_X(\omega;t)$. In terms of these Spectral centroid is expectation $E[\omega]$ and Spectral rolloff is $\arg\min_\omega P_X(\omega;t) \geq 0.85$.

5 Spectral Flux is a squared difference between the normalized magnitudes of successive spectral distributions -

$$\frac{||X(\omega;t) - X(\omega;t-1)||_2}{\sqrt{\frac{F}{2}}},$$

where F is length of Fourier spectrum.

6 Zero-crossings - number of times the time-domain signal amplitude changes sign.

Long-term audio features

1. Octave-based modulation spectral contrast (OMSC) - feature extracted from long-term modulation spectrum analysis to describe the time-varying behavior of the music signals. See [2].

2. Low-energy - the percentage of analysis windows that have less RMS energy than the average RMS energy across the texture window. $RMS(y) = \frac{||y||^2}{N}$

3,4. Modulation spectral flatness measure (MSFM) and Modulation spectral crest measure (MSCM) are modified versions of spectral flatness measure (SFM) and spectral crest measure (SCM) respectively where the SFM and the SCM are the features that represent the short-term spectrum. See [3].

The dimension of feature representation of 30-sec song is **1150**.

3.2 Handcrafted (Echonest)

The Echonest Analyzer (Jehan, 2014) implements an onset detector which is used to localize music events called Segments. These segments are described as sound entities that are relative uniform in timbre and harmony and are the basis for further feature extraction. For each Segment the following features are derived from musical audio signals:

- **Segments Timbre** - 12 dimensional vector with unbounded values centered around 0 representing a high level abstraction of the spectral surface.

- **Segments Pitches** - normalized 12 dimensional vector ranging from 0 to 1 corresponding to the 12 pitch classes C, C#, to B.

- **Segments Loudness Max** represents the peak loudness value within each segment.

- **Segments Loudness Max Time** describes the offset within the segment of the point of maximum loudness.

- **Segments Start** provide start time information of each segment/onset.

Temporal Echonest Features (TEN) [Schindler, 2014]: All statistical moments of Segments Pitches, Segments Timbre, Segments Loudness Max, Segments Loudness Max Time and lengths of segments calculated from Segments Start are calculated.
The dimension of feature representation of 30-sec song is **232**.

3.3 Unsupervised Feature Learning with Sparse Coding

Audio features can be generated automatically under the assumption that audio has a sparse representation (Henaff et al., 2011). To extract learned features each song was first divided into short frames of 1024 samples each, corresponding to 46.4ms of audio. There was a 50% overlap between consecutive frames. Then a Constant-Q transform (CQT) was applied to each frame, with 96 filters spanning four octaves from C2 to C6 at quarter-tone resolution. After that, to make low-energy signals amplified and high-energy ones muted the local contrast normalisation was applied. In the end for the purpose of dictionary learning and sparse coding each frame was normalised to have zero mean and unit L2-norm. We trained dictionary on normalized frames, the algorithm could be described as follows :

Find basis (dictionary) $D = [d_1, d_2, \ldots, d_m] \in \mathbb{R}^{n \times m}$ and coefficient matrix $Z \in \mathbb{R}^{m \times N}$, such that obtained projection is ℓ_2 close to the original data matrix $X \in \mathbb{R}^{n \times N}$ and Z is sparse:

$$\min_{Z,D} \frac{1}{2}||X - DZ||_2^2 + \lambda||Z||_1, \text{ s.t. } ||d_j||_2 \leq 1, \forall j \in [1, \ldots, m].$$

We use $m = 225$ element dictionary, that means that the dimension of feature representation of 30-sec song is **225**.

4 t-SNE as a visualisation technique

t-distributed stochastic neighbor embedding (t-SNE) is a nonlinear dimensionality reduction technique for embedding high-dimensional data into a low-dimensional space (e.g. 2D).
The t-SNE algorithms comprises two main stages:

1. Given a set of N data points $x_1, \ldots, x_N \in \mathbb{R}^n$ ($n \gg 1$), t-SNE computes probabilities p_{ij} that are proportional to the similatrity of data points x_i and x_j:

$$p_{i|j} = \frac{e^{-\frac{||x_i - x_j||^2}{2\sigma_i^2}}}{\sum_{k \neq i} e^{-\frac{||x_i - x_k||^2}{2\sigma_i^2}}},$$

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2N}.$$

2. t-SNE defines a similar probability distribution over the points $y_1, \ldots, y_N \in \mathbb{R}^2$ in the low-dimensional map:

$$q_{ij} = \frac{\sum_{k \neq m}(1 + ||y_k - y_m||^2)}{1 + ||y_i - y_j||^2}$$

and minimizes the Kullback–Leibler divergence between the two distributions:

$$\min_{\{y_i\}_{i=1}^N} KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}.$$

5 Deep Learning

Deep neural network:

- uses a cascade of many layers of nonlinear processing units for feature extraction and transformation. The next layer uses the output from the previous layer as input.
- is based on the unsupervised learning of multiple levels of features or representations of the data. Higher level features are derived from lower level features to form a hierarchical representation.
- learns multiple levels of representations that correspond to different levels of abstraction; the levels form a hierarchy of concepts

Assumption:
Deep Learning should distangle complex and high-dimensional datasets by learning new representations in which classes are more and more discriminated, see figure 2.

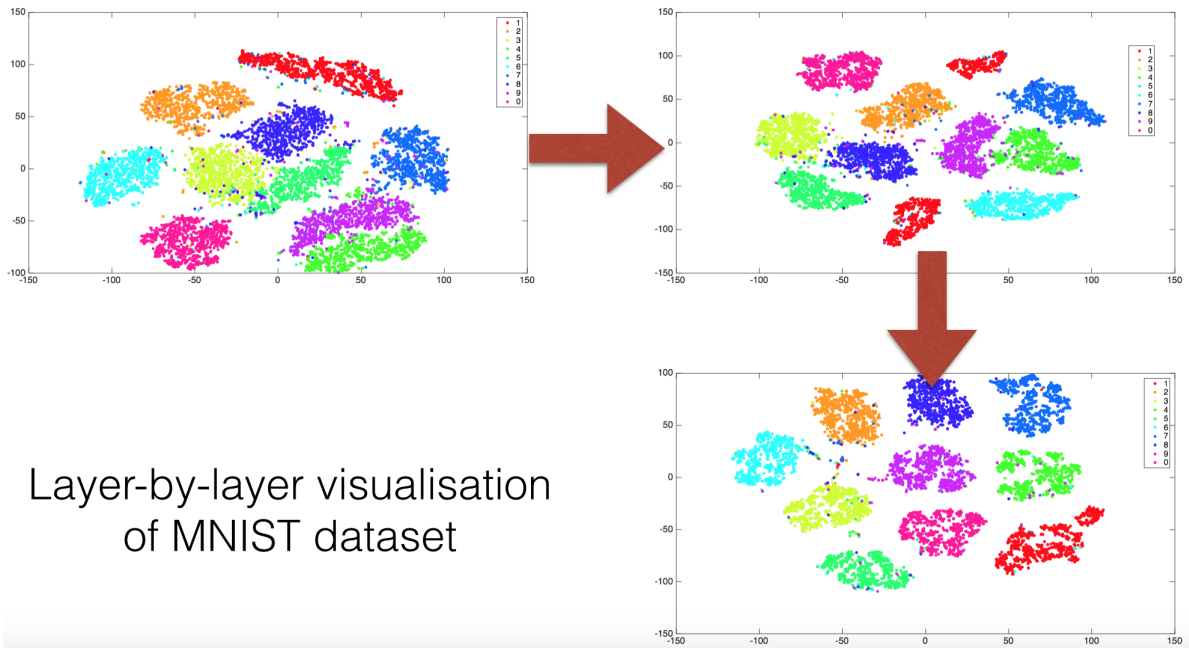

Figure 2: t-SNE visualisation of Deep Learning hierarchical representations of MNIST dataset

6 Datasets description

6.1 MNIST

The MNIST [LeCun et al., 1998] dataset consists of 60,000 images of handwritten digits ('0'-'9'). Image size is 28x28, dimensionality 784. See the t-SNE projection of this dataset in figure 3.

6.2 GTZAN

The GTZAN dataset [Tzanetakis, 2002] consists of 1,000 audio tracks of 30 second length. It contains 10 genres, each represented by 100 tracks. The tracks are all 22050Hz Mono 16-bit audio files in .wav format. See the t-SNE projection of handcrafted feature representation of this dataset in figure 7 and learned feature representation in figure 12.

6.3 MJF

The Montreux Jazz Festival [http://www.montreuxjazz.com/] dataset consists of 9,912 samples. We extracted a subset of 3,726 samples to have 9 balanced classes (genres) with 414 songs in each. We considered for each song its temporal echonest features. See the t-SNE projection of this dataset in figure 15.

7 Data Exploration

7.1 MNIST

For the purpose of experiment we used 200-200-10 neural network architechture (2 hidden layers with 200 neurons each). The classifier based on this architecture has 97.8% accuracy on test set, that means that this model has good generalization and therefore models parameters should contain the structure of the data.
In figure 3 one can see the original MNIST dataset (each element is in $\mathbb{R}^{784}$) projected by t-SNE on 2D plane. One could clearly see that algorithm devided the data into clusters where each corresponds to a different digit. However, the clusters are badly separated, and the margins between some of them are hardly visible. For example group of clusters corresponding to digits '8','3','5' and group '4','9'. Each of these groups looks like one big cluster and their components are hardly visible. This is consistent with human perception, because both groups of handwritten digits are very similar between each other - ('4' and '9') - '9' could be seen as a rounded '4'; ('8','3' and '5') - '8' could be seen as a closed version of '3' and '5'.
In figures 4 and 5 one can see the t-SNE visualization of features extracted by 1^{st} and 2^{nd} hidden layer of NN correspondingly(since architechture is 200-200-10, features are in $\mathbb{R}^{200}$). As one could see, the cluster separability is visually improving with each layer. While the visualization of original data(could be considered as layer 0) has bad cluster separation, the visualization of second-layer features consists of clearly visible and separated 10 clusters. In figure 2 one could notice that t-SNE divided the data into 11 distinct clusters and digit '1' has two clusters. This is explained by two different ways of writing this digit, see figure 6. On the first hidden layer neural network(NN) found 11 types of digits and accrding to NN no other digit except '1' has two sufficiently different ways of writing it.

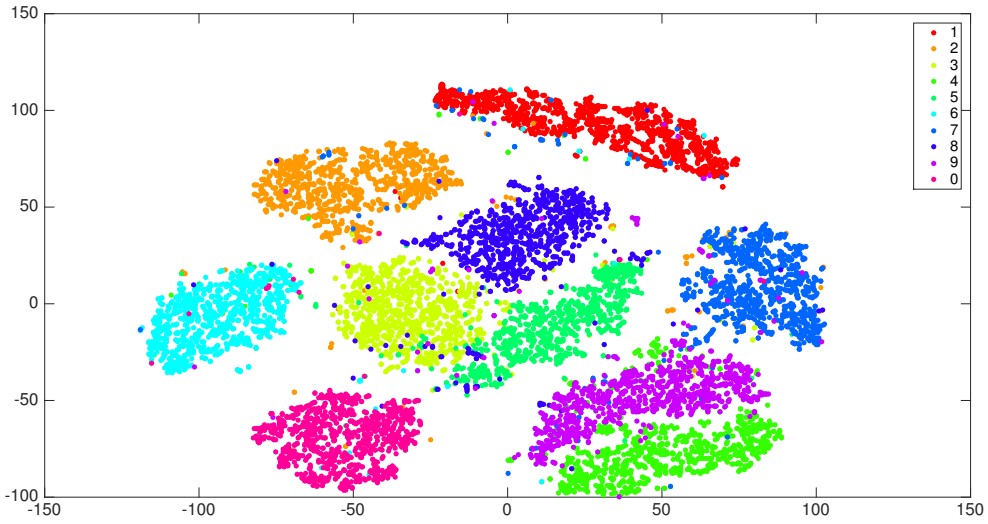

Figure 3: Original MNIST dataset visualized by t-SNE

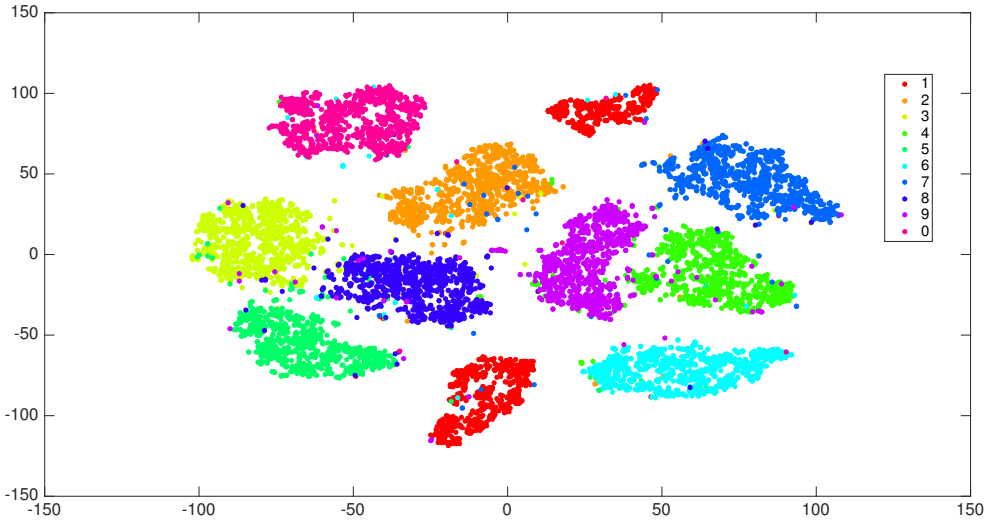

Figure 4: First-layer features of MNIST dataset visualized by t-SNE

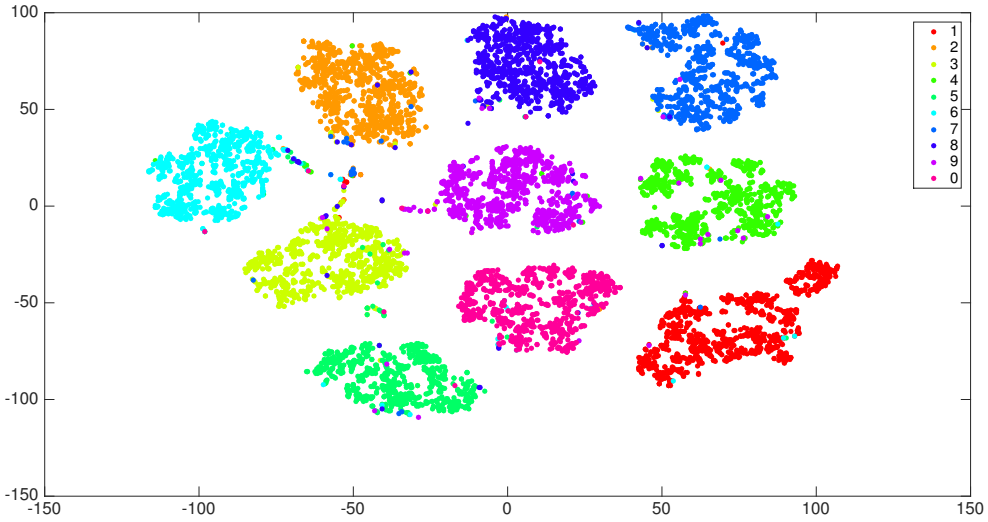

Figure 5: Second-layer features of MNIST dataset visualized by t-SNE

Figure 6: Different ways of writing digit '1'

7.2 GTZAN

7.2.1 Deep Learning + handcrafted

In figure 7 one can see the feature representation of GTZAN dataset projected by t-SNE on 2D plane. One could clearly see, that in opposite to MNIST, the data is badly mixed. The clusters of songs with same genre are not visible. The only visible group is classical music and that proves us that classical music is pure genre and is very different from others.

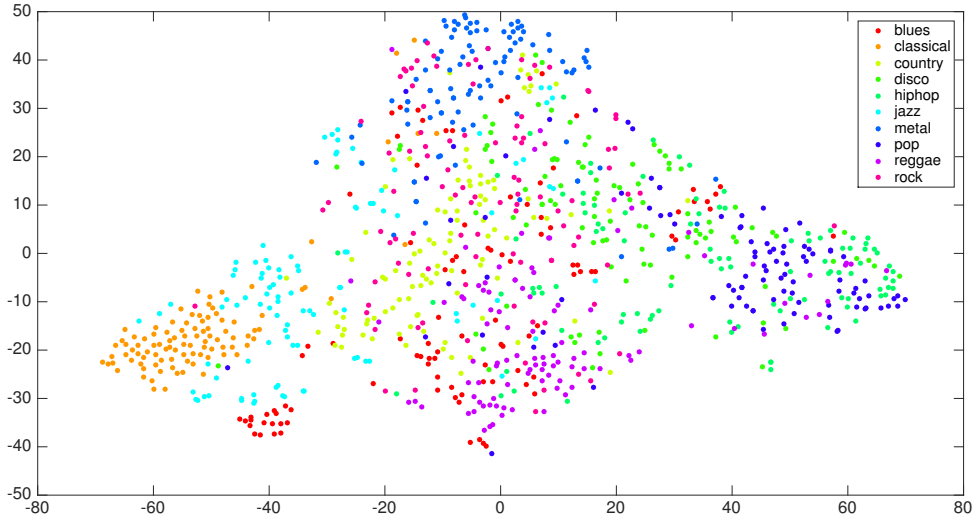

Figure 7: Original GTZAN dataset using [Sturm '13] features visualized by t-SNE

To discriminate the genres so we could see the patterns that are hidden in the projection of the original dataset, we used 200-200-10 neural network. The first layer representation is given in figure 8.

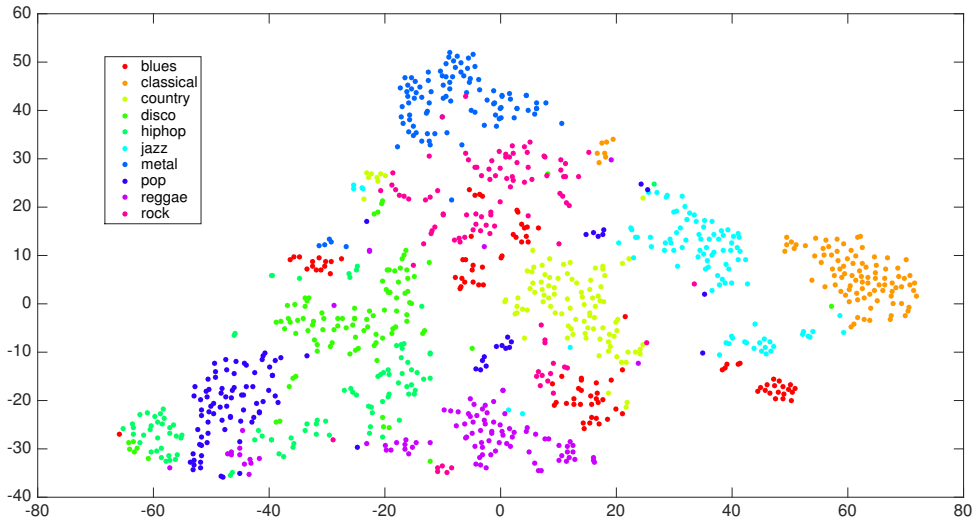

Figure 8: First-layer features of GTZAN dataset

As we can see, the genres are much less mixed and the projection's locations are intuitive to humans - the projection highligted three exteme genres, see figure 9. At the right side is the classical music with jazz and part of blues, at the top is metal with rock close to it and on the left side is active dancing music - disco, pop and hip-hop.

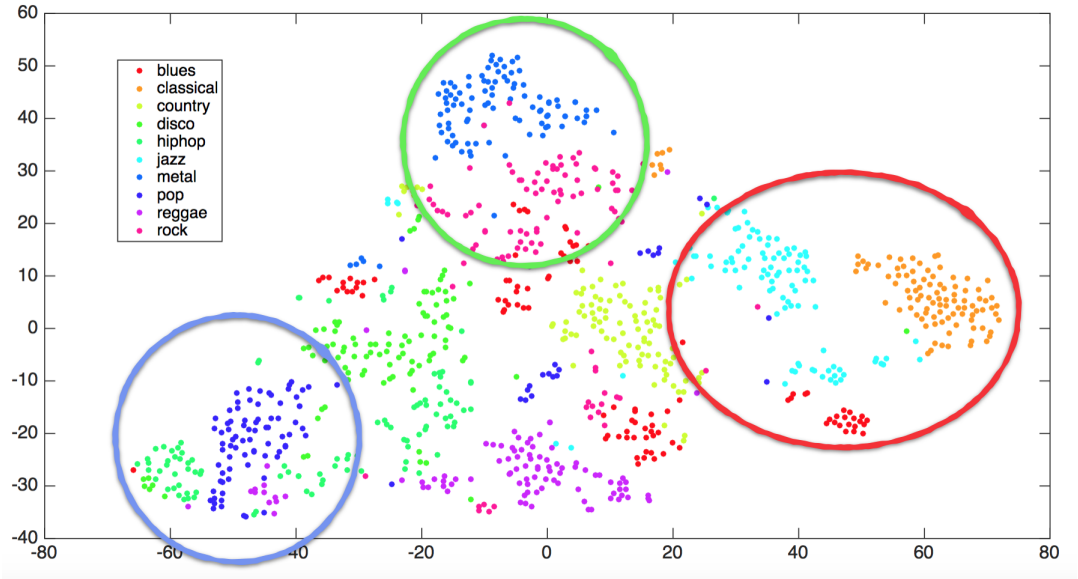

Figure 9: First-layer features of GTZAN dataset, interesting regions are highlighted

The second layer representation is given in figure 10.

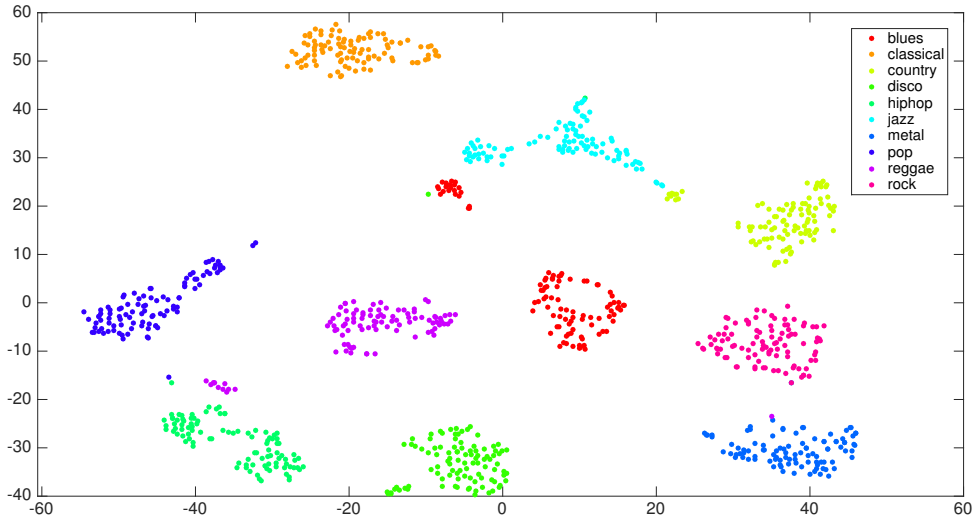

Figure 10: Second-layer features of GTZAN dataset

Figure 11 shows some interesting structural properties of GTZAN dataset. For instance, blues is divided into 2 parts – first part is separated from other genres, while the second lies very close to jazz. This is not surprising since jazz was originated from blues. Same happens with reggie and hip-hop – small group of reggie data lies very close to hip-hop (hip-hop was originated from reggie and funk). Moreover, we see that hip-hop is divided into two clusters, after listening to the songs in each cluster, we found out that the left claster corresponds to "old-school" subgenre and right to the "rap" subgenre.

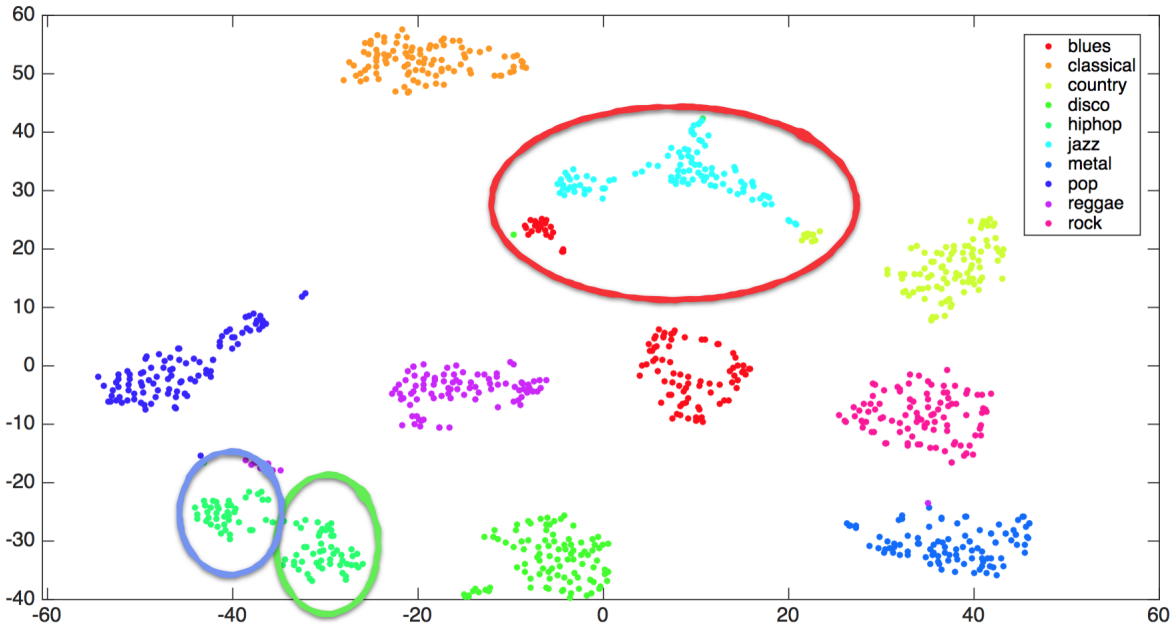

Figure 11: Second-layer features of GTZAN dataset, interesting regions are highlighted

7.2.2 Deep Learning + learned

In figure 12 one can see the t-SNE projection of the original GTZAN dataset in learned feature representation.

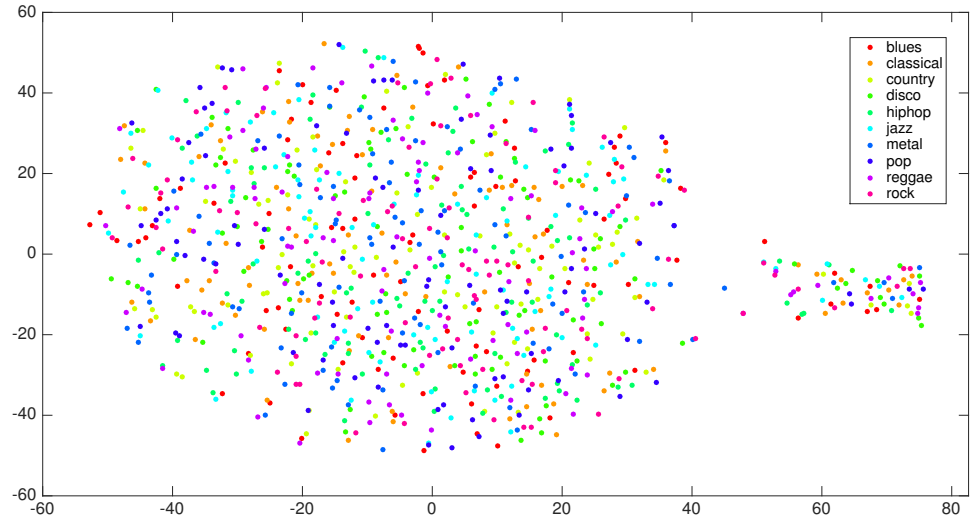

Figure 12: Original GTZAN dataset using learned features visualized by t-SNE

As can be seen the data is highly mixed and the structure is not visible. To pick out the structure we learned 165-165-10 neural network. The representation given by the first NN's hidden layer is shown in figure 13.

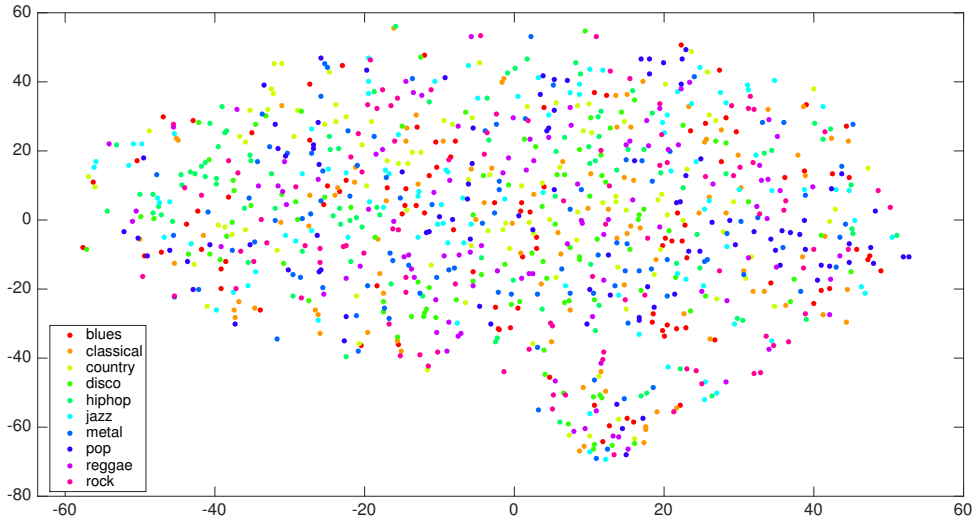

Figure 13: First-layer features of GTZAN dataset

As could be seen there is no improvement, the data is so mixed that we need to go to deeper representations. The second layer representation is given in figure 14.

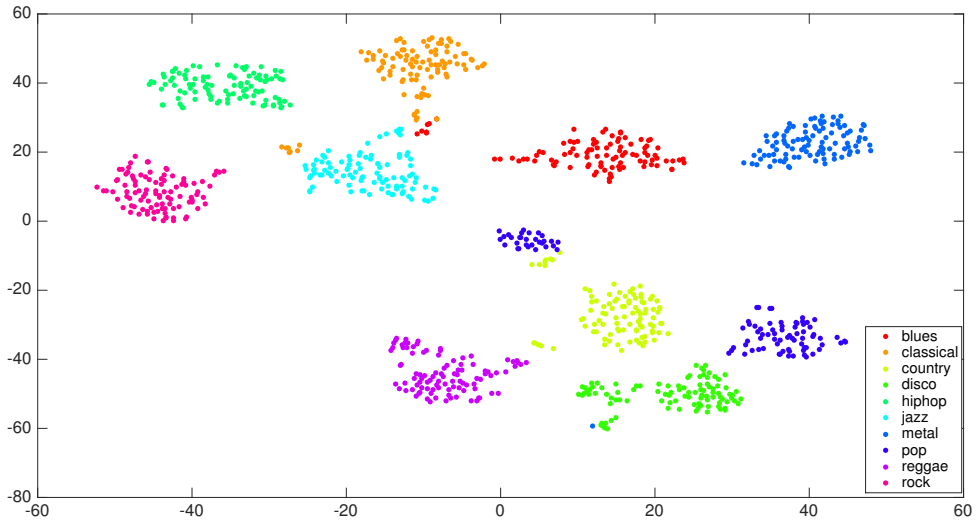

Figure 14: Second-layer features of GTZAN dataset

Now the data is very well separated and this visualisation is consistent with the one obtained from handcrafted features - jazz and part of blues are close to the classic music and hip-hop is divided into two classes.

7.3 MJF

In figure 15 the original echonest feature representation of MJF is shown. As could be seen the data is very mixed, to pick out the structure we used 300-300-300-9 neural network. The first, second and third layer representations are shown in the figures 16, 17 and 18 correspondingly. The genres become more and more discriminated with layer number, however even on the third layer the data is still mixed.

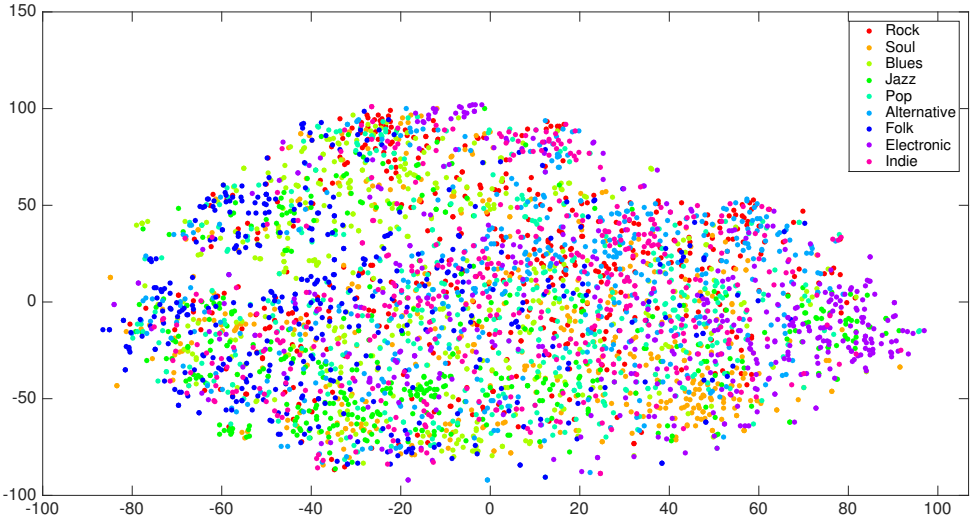

Figure 15: Original MJF dataset using echonest features visualized by t-SNE

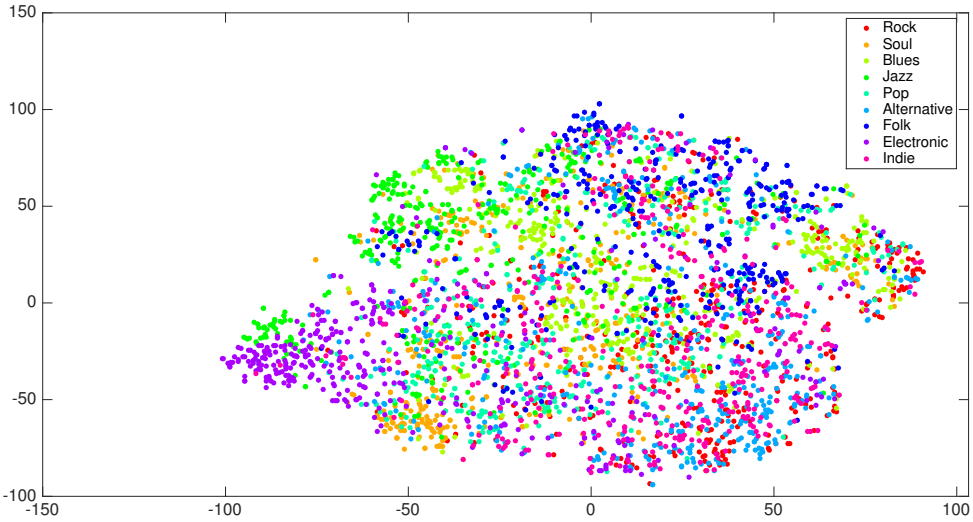

Figure 16: First-layer features of MJF dataset

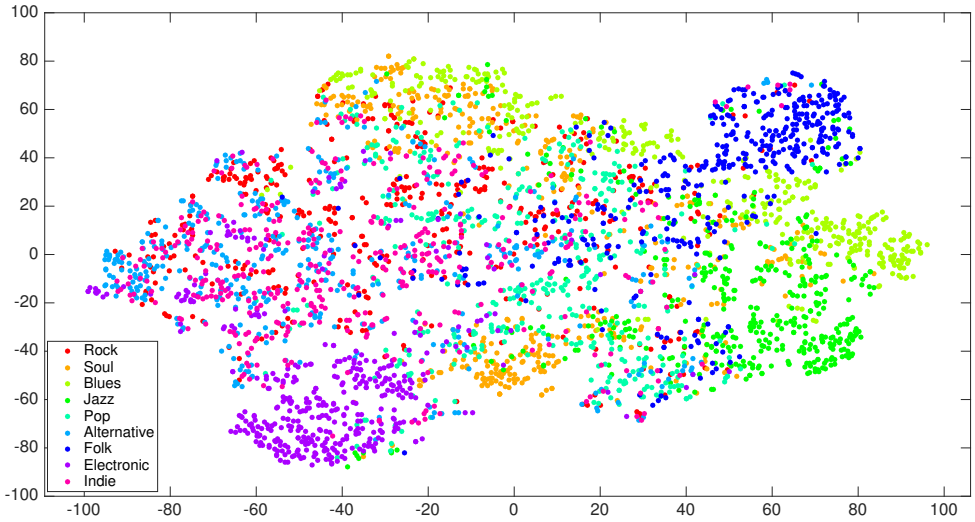

Figure 17: Second-layer features of MJF dataset

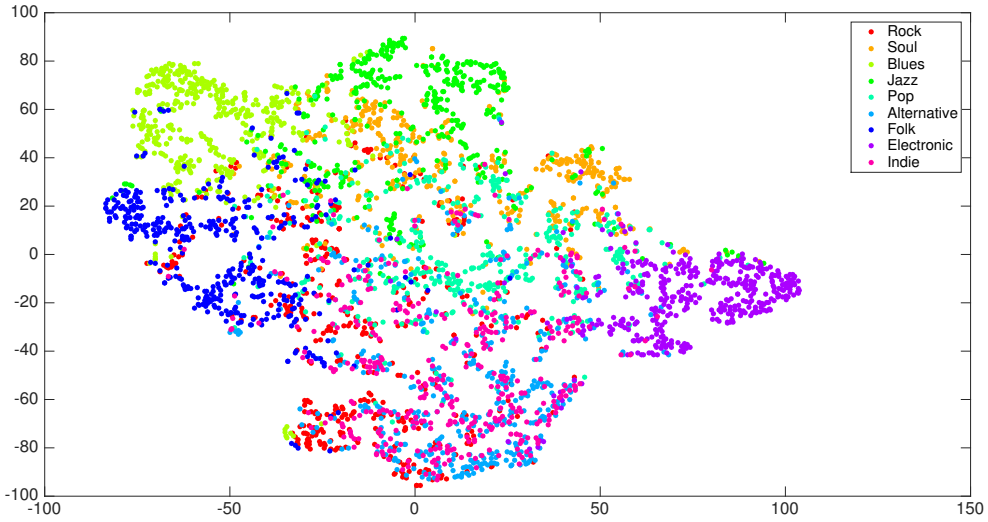

Figure 18: Third-layer features of MJF dataset

As could be seen from the last layer representation Indie and Alternative genres overlap each other, see figure 19. Since that happens even on the third layer, we can conclude that for the algorithm finds those genres similar. And that correspond to the ground truth, because Alternative and Indie are interchangeable terms, the difference is only that Alternative was the preferred American term and Indie came from the United Kingdom.

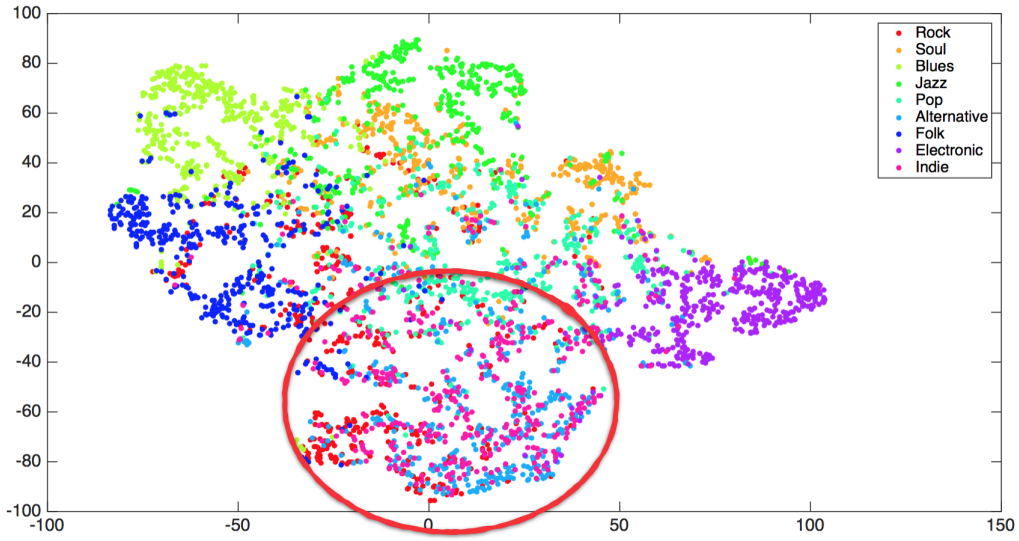

Figure 19: Third-layer features of MJF dataset (Indie and Alternative)

Another interesting property of this dataset is the position of genres Jazz, Blues and Soul, see figure 20. The Soul genre clasterize between Blues and Jazz - that understandable since Soul appeared as a mix of Jazz and Blues.

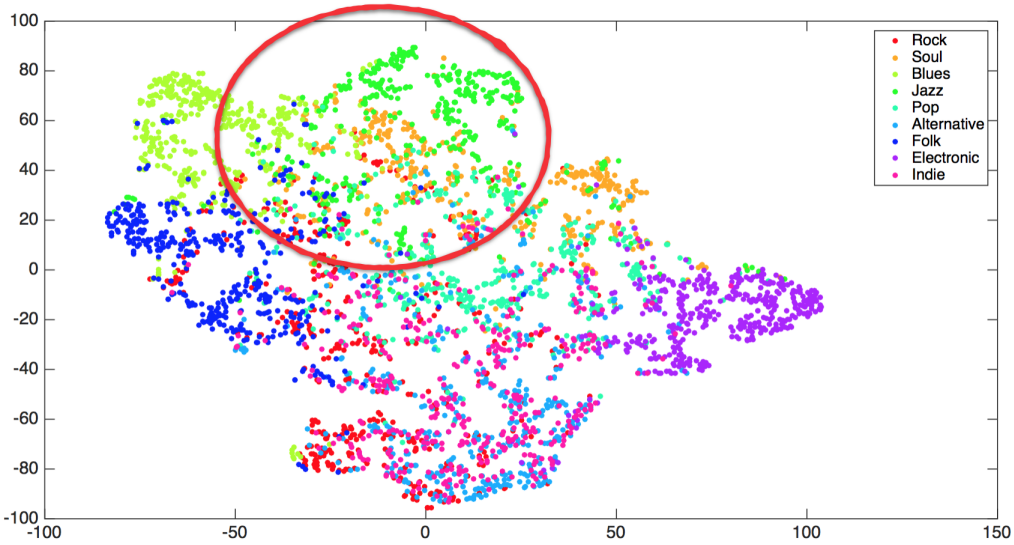

Figure 20: Third-layer features of MJF dataset (Jazz, Blues and Soul)

8 Music recommendation

Since songs that are similar for human perception in the Deep Learning+t-SNE 2D projection are close to each other, the proposed visualisation method could be used to

produce a playlist with smooth transitions between consecutive songs. Here we present two methods of creating the playlist - Shortest Path and Random Walk.

8.1 Shortest path

In this method user choses two songs - the first and the last in the playlist, e.g. the first is Classical and the last is Pop (see figure 21). To produce a playlist with smooth transitions we build a k-nearest neighbour graph based on 2D projection of music dataset. The shortest path between two given points computed by Dijkstra's algorithm form a required playlist.

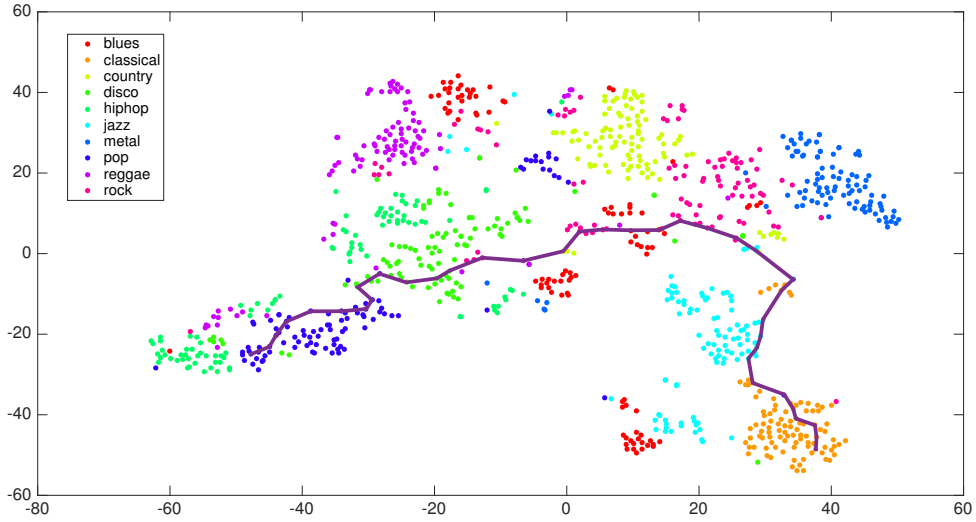

Figure 21: Playlist created by shortest path algorithm

8.2 Random walk

In this method user choses only one song which he wants to listen to first and the playlist is generated by random walk algorithm on the k-nearest neighbour graph from subsection above. The example of random walk playlist is shown in figure 22.

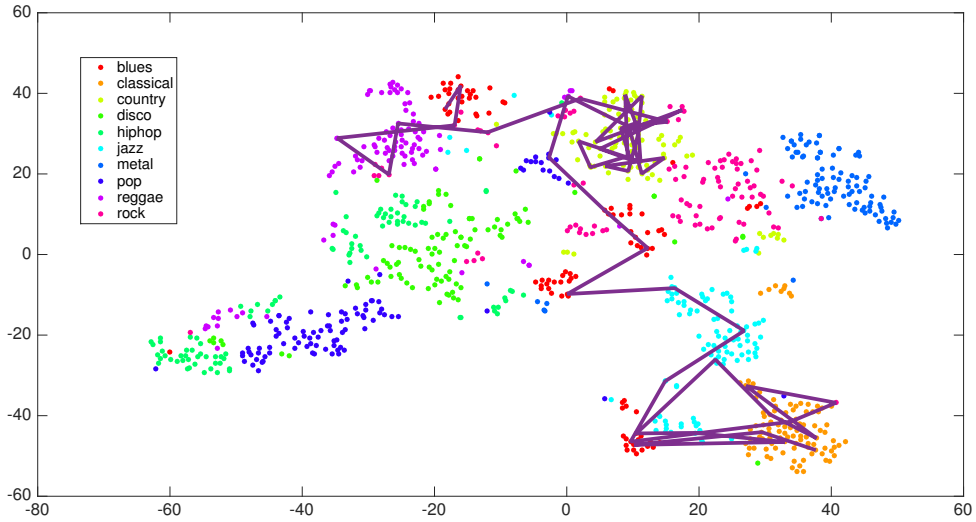

Figure 22: Playlist created by random walk algorithm

9 Conclusion

Deep Learning + t-SNE is a promising visualisation tool for complex high-dimensional datasets. The applications are music recommendation and structure mining. For future development we plan to combine visualisation with Deep Learning, so the user could predefine on which structural properties of the dataset he wants to concentrate and deep learning visualisation would be performed in a way to emphasise those properties.

10 References

1. Sturm, B.L., "On music genre classification via compressive sampling," Multimedia and Expo (ICME), 2013 IEEE International Conference on , vol., no., pp.1,6, 15-19 July 2013

2. Chang-Hsing Lee; Jau-Ling Shih; Kun-Ming Yu; Jung-Mau Su, "Automatic Music Genre Classification using Modulation Spectral Contrast Feature," Multimedia and Expo, 2007 IEEE International Conference on , vol., no., pp.204,207, 2-5 July 2007

3. Dalwon Jang; Minho Jin; Yoo, C.D., "Music genre classification using novel features and a weighted voting method," Multimedia and Expo, 2008 IEEE International Conference on , vol., no., pp.1377,1380, June 23 2008-April 26 2008

4. Jehan, T. and DesRoches, D., "Analyzer documentation", 2014. Available online at `http://developer.echonest.com/docs/v4/_static/AnalyzeDocumentation.pdf`; visited on May 26th 2015

5. Montreux Jazz Festival - `http://www.montreuxjazz.com/`

6. Hinton, G. E., Osindero, S. and Teh, Y., "A fast learning algorithm for deep belief nets," Neural Computation, 18, pp 1527-1554, 2006

7. Tzanetakis, G. and Cook, P., "Musical genre classification of audio signals," IEEE Transactions on Audio and Speech Processing, 2002

8. Schindler, A. and Rauber, A., "Capturing the temporal domain in echonest features for improved classification effectiveness," Adaptive Multimedia Retrieval: Semantics, Context, and Adaptation, pp. 214–227. Springer, 2014

9. van der Maaten, L.J.P. and Hinton, G.E., "Visualizing High-Dimensional Data Using t-SNE," Journal of Machine Learning Research 9(Nov), pp. 2579-2605, 2008.