International Conference on New Interfaces for Musical Expression

Hinged t-SNE for Musical Interfaces

Lilac Atassi

URL: https://nime.pubpub.org/pub/ojkhu4va

Hinged t-SNE for Musical Interfaces

Author keywords

t-SNE, dimensionality reduction, interface

Research questions

t-SNE[1] has multiple desirable properties, including well separation of distant points. The dimensionality reduction algorithm has been used in multiple interfaces to map parameters from a high dimensional space to two or three dimensions. The research question this paper investigates is whether it is possible to preserve desirable properties of t-SNE and control the position of some points in the lower dimensions. This modification turns t-SNE to a more suitable method for a musical interface as the designer can design the layout of the points in the lower dimension presented to the performer.

Two alternative methods are empirically evaluated in this project. In the first method, the selected points are fixed at a position in the lower dimension. These fixed points pull and push the other points as usual in t-SNE but do not move themselves. In the second method, the selected points iteratively move closer to their set destinations, pushing and pulling the other points along the way.

Context

A common problem in musical interface design is presenting a set of high dimensional parameters. High dimension in the context is relative to the dimensionality of the interface. For a two dimensional interface, three and more dimensions need to be mapped onto the lower dimension. The dimensionality of the parameters means the number of scalar parameters for a synthesizer or the dimensionality of a latent vector in a deep generative model, for example. Therefore, this is the general dimensionality reduction problem, mapping high dimensional points (in this context, the parameters) to a lower dimensionality.

Where the number of points is small, such that the corresponding points in the lower dimension can be placed by hand, the interface designer based on some principles can do so. For example, the designer may decide to place parameters that produce similar sounds near each other. Designing the mapping by hand becomes intractable where there are a large number of points. Therefore, an algorithm can be used to design the mapping. t-SNE attempts to keep points close in the higher dimension close in the

lower dimension. This property of t-SNE is desirable for many musical interface as potentially can reduce the cognitive load of the performer learning how the parameters are distributed in the two dimensional interface, for example.

At the performance time, the user selects a point in the lower dimensional space, and with interpolation a point in the higher dimensional space is generated which subsequently is used to generate sound or symbolic music. There are several interpolation methods for this purpose including [6] which has multiple preferable properties.

The possibility of positioning data points freely anywhere on the surface of the interface has been discussed as one of the desirable properties of a mapping [6]. Thus, t-SNE for mappings lacks one of the main preferable properties for interface design. To understand the proposed method here to add this property to t-SNE, a high level understanding of the t-SNE algorithm is needed.

The t-SNE algorithm is presented in Algorithm 1 in [1]. The points in the lower dimensional space are randomly initialized close to zero (the range on each axis in some implantations is about 1e-4 or less). Then the low dimensional points are iteratively updated to minimize the cost function. The cost function is the Kullback–Leibler divergence between the probability distributions constructed over pairs of points in the high dimension and low dimension spaces. The iterative minimization using gradient descent is stopped when the largest update amount by gradient descent is smaller than a threshold for number of iterations or after a maximum number of iterations.

Methods

It is possible to influence the layout of the points in the lower dimension by setting the position of a small subset of points for t-SNE. I propose the name hinged t-SNE for this method as the remaining points are influenced by the fixed points and gradually move until the cost function is minimized with the position of the fixed points as the constraints.

Therefore, the two possible approaches to partially control the layout of the low dimensional points are 1. to set the initial position of some points to the desired position and do not update their position in the t-SNE update iterations, or 2. to modify the iterative update step for some points to gradually move to the desired position. In the second method the update step is

$$p_x^i = \frac{d_x}{n} i,$$

where p_x^i is the position along the x axis at the step i, d_x is the destination position along the x axis, n is the number of update steps, and i is the update step number. Similarly the position of the point along the y axis is updated.

In some of my experiments, with the first method, t-SNE failed to take the similar points close to the fixed point. This is likely because of the momentum term in the gradient descent update step. The large distance between the fixed point and the other points leads to a large momentum value. The large momentum pushes the similar points away from the fixed points, preventing convergence to an optimum.

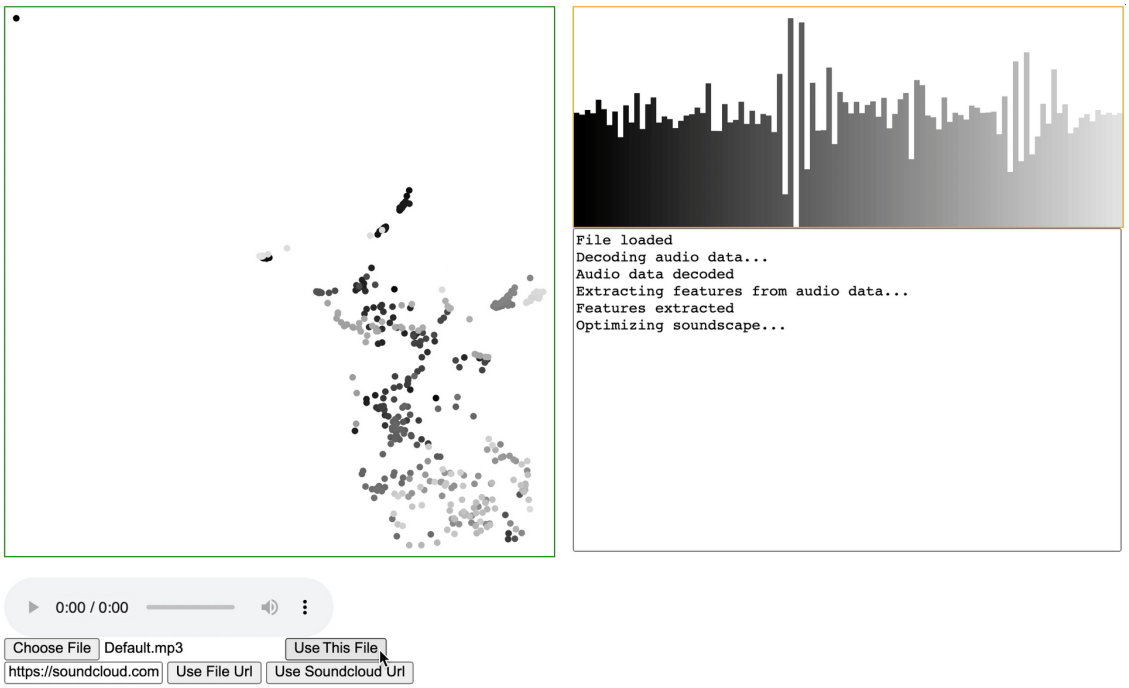
The second method avoids the problem with the disproportionally large momentums. As the fixed point gradually escapes to its final position, its nearby points in the higher dimension have the chance to follow it. In turn, the distance between the fixed point and similar points in the higher dimension does not become much larger than the distance between the pairs of the similar points. Therefore, the gradient descent optimization similar to the original t-SNE algorithm manages to converge to an optimum.

The implementation of both methods is straightforward based on open source t-SNE code. For instance, openTSNE [2] can call a callback function at the end of each optimization iteration. And in the callback function it is possible to override the position of the points.

As an experiment, I implemented Hinged t-SNE in a music remixing interface. MusicMappr [7] splits a given audio file into spectrograms of pre-set length. t-SNE is used to visualize the spectrograms on the screen. One downside of t-SNE in this application is that the points corresponding to the spectrograms end up at random positions. The user needs to spend time to discover the layout of the points. By prepositioning some of the spectrograms points on the 2d screen using Hinged t-SNE, for example, the user would know the first spectrogram and the similar sounding excerpts are always at the top left corner. The modified code replacing t-SNE with Hinged t-SNE in MusicMappr is available in this github repository.

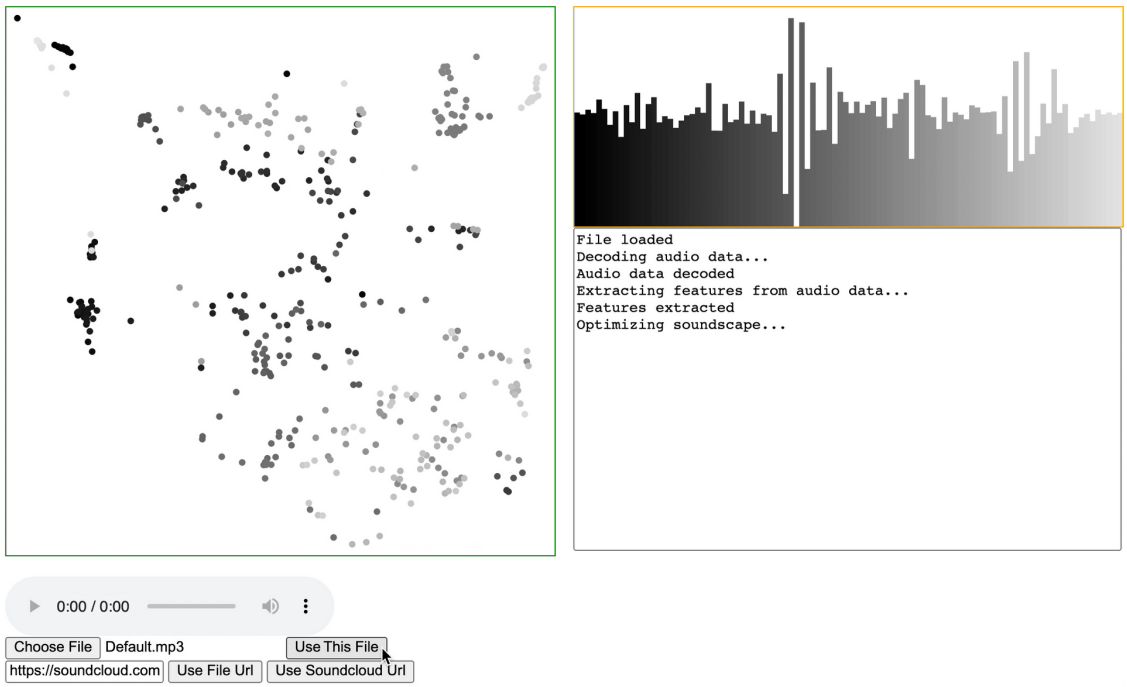
In the following two experiments, the point corresponding to the first spectrogram is fixed at the top left corner of the screen. The view containing the two dimensional points has a dynamic scale. Hence, in the first experiment demonstrating the first

method it appears the fixed point is not stationary while it is. The following picture is a screenshot of the interface with the first Hinged t-SNE method. In the screen recording, one would notice the similar points to the first spectrogram move toward the top left corner and then move away from the fixed point.



The first method is used to pin the first spectrogram point to the top left corner of the interface surface. The other points are being updated by the algorithm half way into the optimization process. The screenshot is linked to the screen recording video on youtube.

In the second experiment, using the second method the first spectrogram point gradually moves to the top left corner of the interface. In the screen recording (see the screenshot below for the link), one would notice the similar points follow the first point and end up close to it. The distribution of the other points is not disturbed by this method.



The second method is used to position the first spectrogram point at the top left corner of the interface surface. The other points are being updated by the algorithm during the optimization process. The screenshot is linked to the screen recording video on youtube.

In conclusion, the proposed Hinged t-SNE with gradual updates of the fixed points (second method) in practice is able to position the selected points freely without impacting the optimization process of t-SNE. This spatial control over the layout of the points in the lower dimension is desirable property for musical interfaces.

Expected outcomes

Among the dimensionality reduction methods for a large set of points, t-SNE appears to be one of the or the most popular choice for music interface researchers (for instance see [3,4,5]). I hope my technical contribution, adding some degree of spatial control to t-SNE, in the Doctoral Consortium gains the attention of the researchers and practitioners who may use the proposed Hinged t-SNE in their future work. Consequently, the feedback from researchers in the DC with substantial experience in interface design for generative models can help guide my future research in this area. Ideally I will be able to collaborate with some practitioners to integrate Hinged t-SNE in some interfaces as well.

Bibliography

[1] Maaten, Laurens van der and Geoffrey E. Hinton. “Visualizing Data using t-SNE.” *Journal of Machine Learning Research* 9 (2008): 2579-2605.

[2] Policar, Pavlin Gregor et al. “openTSNE: a modular Python library for t-SNE dimensionality reduction and embedding.” *bioRxiv* (2019): n. pag.

[3] Lionello, Matteo, et al. "Interactive exploration of musical space with parametric t-SNE." *15th Sound and Music Computing Conference (SMC 2018)*. Sound and Music Computing Network, 2018.

[4] Guedes, Carlos, Konstantinos Trochidis, and Akshay Anantapadmanabhan. "CAMeL: carnatic percussion music generation using N-gram and clustering approaches." *Abstract of Presentation at the 16th Rhythm Production and Perception Workshop, Birmingham*. 2017.

[5] Dupont, Stéphane, et al. "Nonlinear dimensionality reduction approaches applied to music and textural sounds." *2013 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2013.

[6] Marier, Martin. “Designing Mappings for Musical Interfaces Using Preset Interpolation.” *NIME* (2012).

[7] Benjamin, Ethan and Jaan Altosaar. “Musicmapper: interactive 2d representations of music samples for in-browser remixing and exploration.” NIME (2015).

Media

Method 1, the selected point is pinned at the top left corner. The similar points first gravitate to the fixed point but the pass by due to the large momentum. The selected point ends close to non-similar points.

Method 2, the selected point gradually moves to the top left corner. The similar points closely follow the selected point and end up in the same cluster.