A MODEL FOR ADAPTIVE REDUCED-DIMENSIONALITY EQUALISATION

Spyridon Stasis, Ryan Stables, Jason Hockman

DMT Lab,
Birmingham City University,
Birmingham, UK.
spyridon.stasis@bcu.ac.uk
ryan.stables@bcu.ac.uk
jason.hockman@bcu.ac.uk

ABSTRACT

We present a method for mapping between the input space of a parametric equaliser and a lower-dimensional representation, whilst preserving the effect's dependency on the incoming audio signal. The model consists of a parameter weighting stage in which the parameters are scaled to spectral features of the audio signal, followed by a mapping process, in which the equaliser's 13 inputs are converted to (x, y) coordinates. The model is trained with parameter space data representing two timbral adjectives (*warm* and *bright*), measured across a range of musical instrument samples, allowing users to impose a semantically-meaningful timbral modification using the lower-dimensional interface. We test 10 mapping techniques, comprising of dimensionality reduction and reconstruction methods, and show that a stacked autoencoder algorithm exhibits the lowest parameter reconstruction variance, thus providing an accurate map between the input and output space. We demonstrate that the model provides an intuitive method for controlling the audio effect's parameter space, whilst accurately reconstructing the trajectories of each parameter and adapting to the incoming audio spectrum.

1. BACKGROUND

Equalisation is an integral part of the music production workflow, with applications in live sound engineering, recording, music production and mastering, in which multiple frequency dependent gains are applied to the audio signal. Generally the process of equalisation can be categorised under one of the following headings, *corrective equalisation*: in which problematic frequencies are often attenuated in order to prevent issues such as acoustic feedback, and *creative equalisation*: in which the audio spectrum is modified to achieve a desirable timbral transformation. The latter often involves a process of translation between a perceived timbral adjective such as *bright*, *flat* or *sibilant* and an audio effect's input space, by which a music producer must reappropriate a perceptual representation of a timbral transformation as a configuration of multiple parameters in an audio processing module. As music production is an inherently technical process this mapping procedure is not necessarily trivial, and is made more complex by the source-dependent nature of the task.

We propose a system that projects the controls of a parametric equaliser comprising 5 biquad filters arranged in series onto an editable two-dimensional space, allowing the user to manipulate the timbre of an audio signal using an intuitive interface. Whilst the axes of the two-dimensional space are somewhat arbitrary, underlying timbral characteristics are projected onto the space via a training stage using musical semantics data. In addition to this, we propose a signal processing method of adapting the parameter modulation process to the incoming audio data based on feature extraction applied to the long-term average spectrum (LTAS), capable of running in near-real-time. The model is implemented using the SAFE architecture (detailed in [1]), and provided as an extension of the current Semantic Audio Parametric Equaliser,[1] shown in Figure 1.

Figure 1: The extended Semantic Audio Equalisation plug-in with the two-dimensional interface. To modify the brightness/warmth of an audio signal, a point is positioned in two-dimensional space.

1.1. Semantic Music Production

Engineers and producers generally use a wide variety of timbral adjectives to describe sound, each with varying levels of agreement. By modelling these adjectives, we are able to provide perceptually meaningful abstractions, which lead to a deeper understanding of musical timbre and system that facilitate the process of audio manipulation. The extent to which timbral adjectives can be accurately modelled is defined by the level of exhibited agreement, a concept investigated in [2], in which terms such as bright, resonant and harsh all exhibit strong agreement scores and terms such as open, hard and heavy all show low subjective agreement scores. It is common for timbral descriptors to be represented in low-dimensional space, brightness for example is shown to exhibit a strong correlation with spectral centroid [3, 4] and has further dependency on the fundamental frequency of the signal [5]. Simi-

[1] Available for download at http://www.semanticaudio.co.uk

larly, studies such as [6] and [7] demonstrate the ability to reduce complex data to lower-dimensional spaces using dimensionality reduction.

Recent studies have also focused on the modification of the audio signal using specific timbral adjectives, where techniques such as spectral morphing [8] and additive synthesis [9] have been applied. For the purposes of equalisation, timbral modification has also been implemented via semantically-meaningful controls and intuitive parameter spaces. SocialEQ [10] for example, collects timbral adjective data via a web interface and approximates the configuration of a graphic equaliser curve using multiple linear regression. Similarly, subjEQt [11], provides a two-dimensional interface, created using a Self Organising Map, in which users can navigate between presets such as *boomy*, *warm* and *edgy* using natural neighbour interpolation. This is a similar model to 2DEQ [12], in which timbral descriptors are projected onto a two-dimensional space using Principal Component Analysis (PCA). The Semantic Audio Feature Extraction (SAFE) project provides a similar non-parametric interface for semantically controlling a suite of audio plug-ins, in which semantics data is collected within the DAW. Adaptive presets can then be selectively derived, based on audio features, parameter data and music production metadata.

2. METHODOLOGY

In order to model the desired relationship between the two parameter spaces, a number of problems must be addressed. Firstly, the data reduction process should account for maximal variance in high-dimensional space, without bias towards a smaller subset of the EQ's parameters. Similarly, we should be able to map to the high-dimensional space with minimal reconstruction error, given a new set of (x, y) coordinates. This process of mapping between spaces is nontrivial, due to loss of information in the reconstruction process. Furthermore, the low-dimensional parameter space should be configured in a way that preserves an underlying timbral characteristic in the data, thus allowing a user to transform the incoming audio signal in a musically meaningful way. Finally, the process of parameter space modification should not be agnostic of the incoming audio signal, meaning any mapping between the two-dimensional plane and the EQ's parameter space should be expressed as a function of the (x, y) coordinates and some representation of the signal's spectral energy. In addition to this, the system should be capable of running in near-real time, enabling its use in a Digital Audio Workstation (DAW) environment.

For addressing these problems, we develop a model that consists of two phases, where the first comprises a training phase, in which a map is derived from a corpus of parameter data, and the second comprises a testing phase in which a user can present (x, y) coordinates and an audio spectrum, resulting in a 13 dimensional vector of parameter state variables. To optimise the mapping process, we experiment with a combination of 3 dimensionality reduction techniques and 3 reconstruction methods, followed by a stacked-autoencoder model that encapsulates both the dimensionality reduction and reconstruction processes. With the purpose of scaling the parameters to the incoming audio signal, we derive a series of weights based on a selection of features, extracted from the signal's LTAS coefficients. To evaluate the model's performance, we train it with binary musical semantics data and measure the parameter-wise reconstruction error, along with inter-class variance in low-dimensional space.

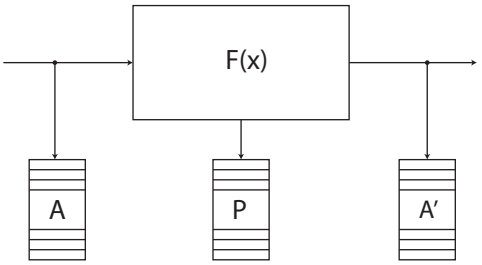


Figure 2: An overview of the SAFE data collection architecture, where A represents the audio features captured before the effect is applied, A' represents the features captured after the effect is applied, and P represents the parameter vector.

2.1. Dataset

For the training of the model, we compile a dataset of 800 semantically annotated EQ parameter space settings, comprising 40 participants equalising 10 musical instrument samples using 2 descriptive terms: *warm* and *bright*. To do this, participants were presented with the musical instrument samples in a DAW and asked to use a parametric equaliser to achieve the two timbral settings. After each setting was recorded, the data were recorded and the equaliser was reset to unity gain. During the test samples were presented to the participants in a random order, across separate DAW channels. Furthermore, the musical instrument samples were all performed unaccompanied, were RMS normalised and ranged from 20 to 30 seconds in length. All of the participants had normal hearing, aged 18-40 and all had at least 3 years' music production experience.

The descriptive terms (*warm* and *bright*) were selected for a number of reasons, firstly the agreement levels exhibited by participants tend to be the high (as suggested by [2]), meaning there should be less intra-class variance when subjectively assigning parameter settings. When measured using an agreement metric, defined by [10] as the log number of terms over the trace of the covariance matrix, *warm* and *bright* were the two highest ranked terms in a dataset of 210 unique adjectives. Secondly, the two terms are deemed to be sufficiently different enough to form an audible timbral variation in low dimensional space. Whilst the two terms do not necessarily exhibit orthogonality (for example brightness can be modified with constant *warmth* [8]), they have relatively dissimilar timbral profiles, with brightness widely accepted to be highly correlated with the signal's spectral centroid, and *warmth* often attributed to the ratio of the first 3 harmonics to the remaining harmonic partials in the magnitude LTAS [13].

The parameter settings were collected using a modified build of the SAFE data collection architecture, in which descriptive terms, audio feature data, parameter data and metadata can be collected remotely, within the DAW environment and uploaded to a server. As illustrated in Figure 2, the SAFE architecture allows for the capture of audio feature data before and after processing has been applied. Similarly, the interface parameters P (see Table 1) are captured and stored in a linked database. For the purpose of this experiment, the architecture was modified by adding the functionality to capture LTAS coefficients, with a window size of 1024 samples and a hop size of 256.

Whilst the SAFE project comprises a number of DAW plug-ins, we focus solely on the parametric equaliser, which utilises 5

biquad filters arranged in series, consisting of a low-shelving filter (LS), 3 peaking filters (Pf_n) and a high-shelving filter (HS), where the LS and HS filters each have two parameters and the (Pf_n) filters each have 3, as described in Table 1.

n	Assignment	Range	n	Assignment	Range
0	LS gain	-12 - 12 dB	7	Pf_1 Q	0.1 - 10 Hz
1	LS Freq	22 - 1,000 Hz	8	Pf_2 Gain	-12 - 12 dB
2	Pf_0 Gain	-12 - 12 dB	9	Pf_2 Freq	220 - 10,000 Hz
3	Pf_0 Freq	82 - 3,900 Hz	10	Pf_2 Q	0.1 - 10 Hz
4	Pf_0 Q	0.1 - 10 Hz	11	HS Gain	-12 - 12 dB
5	Pf_1 Gain	-12 - 12 dB	12	HS Freq	580 - 20,000 Hz
6	Pf_1 Freq	180 - 4,700 Hz			

Table 1: A list of the parameter space variables and their ranges of possible values, taken from the SAFE parametric EQ interface.

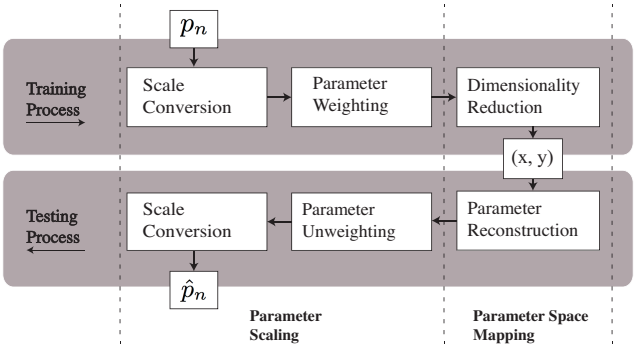


Figure 3: An overview of the proposed model, where the grey horizontal paths represent training and testing phases.

3. MODEL

The proposed system maps between the EQ's parameter space, consisting of 13 filter parameters and a two-dimensional plane, whilst preserving the context-dependent nature of the audio effect. After an initial training phase, the user can then submit (x, y) coordinates to the system using a track-pad interface, resulting in a timbral modification via the corresponding filter parameters. To demonstrate this, we train the model with 2 class (*bright*, *warm*) musical semantics data taken from the SAFE EQ database, thus resulting in an underlying transition between opposing timbral descriptors, in two-dimensional space. By training the model in this manner, we intend to maximise the separability between classes when projected onto the reduced-dimensionality interface.

The model (depicted in Figure 3) has 2 key operations, The first involves weighting the parameters by computing the vector $\alpha_n(A)$ from the input signal's long-term spectral energy (A). We can then modify the parameter vector (P), to obtain a weighted vector (P'). The second component scales the dimensionality of (P'), resulting in a compact, audio-dependent representation. During the model's testing phase, we apply an unweighting procedure, based on the (x, y) coordinates and the signal's modified spectrum. This is done by multiplying the estimated parameters with the inverse weight vector, resulting in an approximation of the original parameters. In addition to the weighting and dimensionality reduction stages, a scale-normalisation procedure is applied aiming to convert the ranges of each parameter (given in Table 1), to ($0 < p_n < 1$). This converts the data into a suitable format for dimensionality reduction.

3.1. Parameter Scaling

As the configuration of the filter parameters assigned to each descriptor by the user during equalisation is likely to vary based on the audio signal being processed, the first requirement of the model is to apply weights to the parameters, based on knowledge of the audio data at the time of processing. To do this, we selectively extract features from the signal's LTAS, before and after the filter is applied. This is possible due to the configuration of the data collection architecture, highlighted in Figure 2. The weights (α_m) can then be expressed as a function of the LTAS, where the function's definition varies based on the parameter's representation (i.e. gain, centre frequency or bandwidth of the corresponding filter). We use the LTAS to prevent the parameters from adapting each time a new frame is read. In practice, we are able to do this by presenting users with means to store the audio data, rather than continually extracting it from the audio stream. Each weighting is defined as the ratio between a spectral feature, taken from the filtered audio signal (A'_k) and the signal filtered by an enclosing rectangular window (R). Here, the rectangular window is bounded by the minimum and maximum frequency values attainable by the observed filter $f_k(A)$

We can define the equaliser as an array of biquad functions arranged in series, as depicted in Eq 1

$$\begin{gathered} f_k = f_{k-1}(A, \vec{P}_{k-1}) \\ k = 1, \ldots, K-1 \end{gathered} \tag{1}$$

Here, $K = 5$ represents the number of filters used by the equaliser and f_k represents the k^{th} biquad function, which we can define by its transfer function, given in Eq 2.

$$H_n(z) = c \cdot \frac{1 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{2}$$

The LTAS is then modified by the filter as in Eq. 3 and the weighted parameter vector can be derived using the function expressed in Eq. 4.

$$A'_k = H_k(e^{j\omega}) A_k \tag{3}$$

$$p'_n = \alpha_m(k) \cdot p_n \tag{4}$$

Where p_n is the n^{th} parameter in the vector P. The weighting function is then defined by the parameter type (m), where $m = 0$ represents gain, $m = 1$ represents centre-frequency and $m = 2$ represents bandwidth. For gain parameters, the weights are expressed as a ratio of the spectral energy in the filtered spectrum (A') to the spectral energy in the enclosing rectangular window (R_n), derived in Eq. 5 and illustrated in Figure 4.

$$\alpha_0(k) = \frac{\sum_i (A'_k)_i}{\sum_i (R_k)_i} \tag{5}$$

For frequency parameters ($m = 1$), the weights are expressed as a ratio of the respective spectral centroids of A' and R_n, as demonstrated in Eq. 6, where b_i are the corresponding frequency bins.

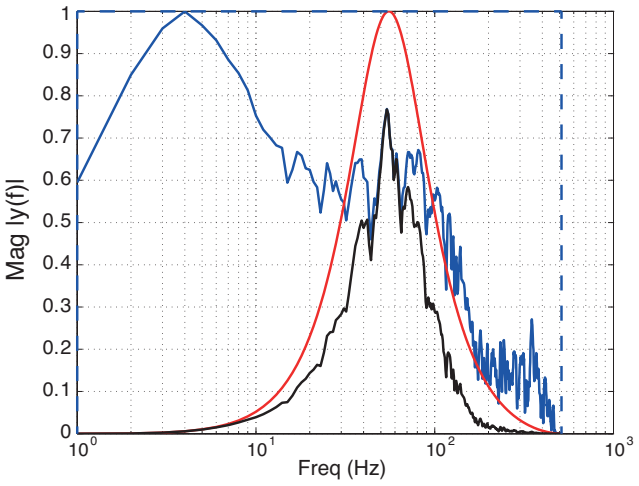


Figure 4: An example spectrum taken from an input example, weighted by the biquad coefficients, where the red line represents a peaking filter, the black line represents the biquad-filtered spectrum and the blue line represents the spectral energy in the rectangular window (R_m).

$$\alpha_1(k) = \left(\frac{\sum_i (A'_k)_i b_i}{\sum_i (A'_k)_i}\right) / \left(\frac{\sum_i (R_k)_i b_i}{\sum_i (R_k)_i}\right) \tag{6}$$

Finally, the weights for bandwidth parameters ($m = 2$) are defined as the ratio of spectral spread exhibited by both A' and R_n. This is demonstrated in Eq. 7, where $(x)_{sc}$ represents the spectral centroid of x.

$$\alpha_2(k) = \left(\frac{\sum_i \left(b_i - (A'_k)_{sc}\right)^2 (A'_k)_i}{\sum_i (A'_k)_i}\right) / \left(\frac{\sum_i \left(b_i - (R_k)_{sc}\right)^2 (R_k)_i}{\sum_i (R_k)_i}\right) \tag{7}$$

During the testing phase, retrieval of the unweighted parameters, given a weighted vector can be achieved by simply multiplying the weighted parameters with the inverse weights vector, as in Eq. 8.

$$\hat{p}_n = \alpha_m^{-1}(k) \cdot \hat{p}'_n \tag{8}$$

Where $\hat{p}$ is a reconstructed version of p, after dimensionality reduction has been applied.

To ensure the parameters are in a consistent format for each of the dimensionality scaling algorithms, a scale normalisation procedure is applied using Eq. 9, where during the training process, the p_{min} and p_{max} represent the minimum and maximum values for each parameter (given in Table 1), and q_{min} and q_{max} represent 0 and 1. During the testing process, these values are exchanged, such that q_{min} and q_{max} represent the minimum and maximum values for each parameter and p_{min} and p_{max} represent 0 and 1.

$$\rho_n = \frac{(p_n - q_{min})(p_{max} - p_{min})}{q_{max} - q_{min}} + p_{min} \tag{9}$$

Additionally, a sorting algorithm was used to place the three mid-band filters in ascending order based on their centre frequency. This prevents normalisation errors due to the frequency ranges allowing the filters to be rearranged by the user.

3.2. Parameter Space Mapping

Once the filters have been weighted by the audio signal, the mapping from 13 EQ variables to a two-dimensional subspace can be accomplished using a range of dimensionality reduction techniques. We start by evaluating the performance of three commonly used algorithms for data reduction by training them with weighted parameter space data and measuring the variance. Conversely, the reconstruction process is less common due to the nature of dimensionality reduction. We evaluate the efficacy of three multivariate regression-based techniques at mapping two-dimensional interface variables to a vector of EQ parameters. This is done by approximating functions using the weighted parameter data and measuring the reconstruction error. Finally, we evaluate a stacked autoencoder model of data reduction, in which the parameter space is both reduced and reconstructed in the same algorithm, we are then able to detach the reconstruction (decoder) stage for the testing process.

Dimensionality reduction is implemented using the following techniques: *Principal Component Analysis* (PCA), a widely used method of embedding data into a linear subspace of reduced dimensionality, by finding the eigenvectors of the covariance matrix, originally proposed by [14]; *Linear Discriminant Analysis* (LDA), a supervised projection technique that maps to a linear subspace whilst maximising the separability between data points that belong to different classes (see [15]); *Kernel PCA* (kPCA), a non-linear manifold mapping technique, in which eigenvectors are computed from a kernel matrix as opposed to the covariance matrix, as defined by [16]. As LDA projects the data-points onto the dimensions that maximise inter-class variance for C classes, the dimensionality of the subspace is set to $C - 1$. This means that in a binary classification problem, such as ours, we need to reconstruct the second dimension arbitrarily. For each of the other algorithms, we select the first 2 variables for mapping, and for the kPCA algorithm, the feature distances are computed using a Gaussian kernel.

The parameter reconstruction process was implemented using the following techniques: *Linear Regression* (LR), a process by which a linear function is used to estimate latent variables; *Natural Neighbour Interpolation* (NNI), a method for interpolating between scattered data points using Voronoi tessellation, as used by [11] for a similar application; *Support Vector Regression* (SVR), a non-linear kernel-based regression technique (see [17]), for which we choose a Gaussian kernel function.

An autoencoder is an Artificial Neural Network with a topology capable of learning a compact representation of a dataset by optimising a matrix of weights, such that a loss function representing the difference between the output and input vectors is minimised. Autoencoders can then be cascaded to form a network and initialised using layer-wise pre-training, commonly using Restricted Boltzmann Machines (RBMs), followed by backpropagation, leading to a complex nonlinear mapping between parameter spaces. This approach has proven to be successful for data compression [18] due to its ability to reconstruct high-dimensional data via a reduced feature subset. In our model, the symmetrical network consists of three hidden layers, with 13 units in the input and output layers, and two units in the central layer. The remaining hidden layers have nine units, and sigmoidal activation functions were used for each node. After the weights were pretrained with the RBMs, further optimisation was performed using back propagation and stochastic mini-batch gradient decent, with a batch size of 10 and a learning rate of 0.1.

P:	0	1	2	3	4	5	6	7	8	9	10	11	12	μ	σ
PCA-LR	0.676	0.180	0.186	0.101	0.036	0.530	0.184	0.024	0.283	0.148	0.023	0.501	0.108	0.229	0.040
LDA-LR	0.229	0.086	0.270	0.061	0.045	0.207	0.117	0.031	0.306	0.097	0.041	0.356	0.124	0.151	0.011
kPCA-LR	0.135	0.069	0.149	0.048	0.041	0.152	0.081	0.028	0.137	0.084	0.030	0.127	0.110	0.091	0.002
PCA-SVR	0.460	0.245	0.262	0.179	0.036	0.433	0.350	0.031	0.403	0.249	0.028	0.365	0.109	0.242	0.024
LDA-SVR	0.219	0.136	0.272	0.102	0.043	0.250	0.174	0.040	0.281	0.102	0.037	0.317	0.126	0.162	0.009
kPCA-SVR	0.126	0.063	0.151	0.047	0.038	0.149	0.075	0.027	0.144	0.088	0.031	0.131	0.104	0.090	0.002
PCA-NNI	0.515	0.354	0.294	0.597	0.028	0.504	0.527	0.052	0.374	0.591	0.024	0.488	0.406	0.366	0.044
LDA-NNI	0.391	0.363	0.333	0.200	0.051	0.314	0.305	0.097	0.294	0.222	0.066	0.396	0.188	0.248	0.015
kPCA-NNI	0.132	0.075	0.168	0.051	0.038	0.170	0.087	0.025	0.154	0.109	0.034	0.132	0.107	0.099	0.002
SAe	0.074	0.077	0.129	0.058	0.045	0.168	0.082	0.022	0.128	0.103	0.027	0.094	0.115	0.086	0.002

Table 2: Mean reconstruction error per parameter using combinations of dimensionality reduction and reconstruction techniques. The final two columns show mean (μ) and variance (σ) per parameter across all techniques. The model with the lowest mean reconstruction error (Stacked Autoencoder) is highlighted in grey.

4. RESULTS AND DISCUSSION

Parameter reconstruction error: We measure the reconstruction error for each combination of dimensionality reduction and reconstruction techniques by computing the mean squared error between predicted and actual parameter states, across all training examples. To do this, we use *K*-fold cross validation with $k = 100$ iterations, and a test partition size of 10% (80 training examples). The mean error for each technique is then calculated and the variance is found per-parameter, as shown in Table 2. As highlighted in the table, the stacked autoencoder model outperforms all combinations of reduction and reconstruction algorithms, with a mean reconstruction error of 0.086. Similarly, the autoencoder equals the lowest parameter variance measurement from all of the evaluated models, showing the consistency in high-dimensional parameter reconstruction. This is a desirable characteristic as it demonstrates that the problem of loading-bias towards a smaller subset of parameters does not exist in the system, which may cause unresponsive filter parameters during the testing phase.

Class Separation: To measure the extent to which the class separation (*warm* and *bright*) has been preserved in low-dimensional space, we measure the 2 sided Kullback-Leibler Divergence (KLD) between classes, after each of the dimensionality reduction techniques have been applied to the dataset. This allows us the measure the relative entropy between the class-distributions, in two-dimensional space. The KLD measurements show the LDA technique exhibited the highest degree of separation with a score of 1.98, whilst the autoencoder performed similarly, with score of 1.63. Conversely, the PCA-based techniques performed less favourably, with kPCA and PCA exhibiting 1.08 and 1.03 respectively. The class-labelled two-dimensional spaces are shown in Figure 5, along with the dimensions of maximal variance and class centroids.

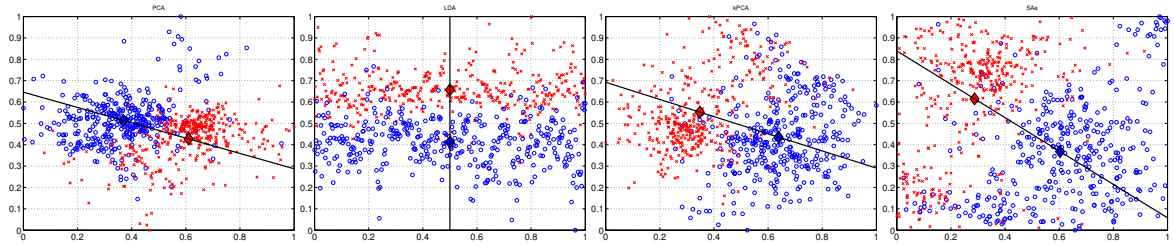
As LDA is a supervised technique, with the sole purpose of preserving discriminatory dimensions, thus maximising the variance between cluster centroids, it is expected that the class discrimination outperforms the other evaluated models. The autoencoder however performs similarly using an unsupervised algorithm, with no prior introduction to class labels. As LDA projects out 2 class data onto a 1 dimensional plane, a 2 class problem such as this (*warm* and *bright*) lacks the additional interface variability. This is shown in Figure 5b, where the spacing of points along the x-axis is arbitrarily based upon their sequential order. The dimension of maximal variance in this case is often aligned vertically, however this can be re-orientated by adding bias values to the data.

Parameter Weighting: To evaluate the effectiveness of the signal-specific weights, we measure the cross entropy of each dimensionality reduction technique before and after the weights have been applied. By doing so, we are able to observe the behaviour of the (*warm* and *bright*) classes in adaptive and non-adaptive reduced dimensionality space. The results show that three of the four techniques exhibit a positive change after signal weighting, with the autoencoder being the highest at +24%, followed by LDA at +19% and PCA at +6%, whereas the cross entropy of the kPCA technique exhibited a small negative change at -4%. Overall the separability of the data were shown to increase with a mean KLD improvement of 11%, suggesting the weighting process is a valuable attribute in the model. This reinforces the views of [4] and [5], who both suggest perceived timbral features such as *brightness* and *warmth* vary with attributes of the input signal such as $f0$ and perceived loudness.

4.1. Discussion

The results suggest that the stacked autoencoder method of reducing and reconstructing the parameter space provides both high reconstruction accuracy and class preservation in low-dimensional space. To implement the equaliser, the auto encoder's weighting matrix is optimised using the musical semantics data, and the decoder weights ($W' \in \mathbb{R}^{<13\times2>}$) are applied to new input data, using $f(W_n^T X + b_n)$, where f is a non-linear activation function and b is an intercept term. This maps a two-dimensional input vector to a 13 dimensional parameter vector, which can then be scaled to the audio spectrum using the inverse weights vector and scale-normalised, as shown in the testing phase of Figure 3.

Whilst the parameter reconstruction of the autoencoder is sufficiently accurate for our application, it is bound by the intrinsic dimensionality of the data, defined as the minimum number of variables required to accurately represent the variance in lower dimensional space. For our *bright*/*warm* parameter-space data, we can show that this intrinsic dimensionality requires three variables, when computed using Maximum Likelihood Estimation as defined by [19]. As our application requires a two-dimensional interface, this means the reconstruction accuracy is limited. To demonstrate this further, the reconstruction accuracy increases to 99.15% when the same autoencoder model is cross-validated with three variables in the hidden layer. This intrinsic dimensionality often dictates the variable reduction process, such as in [18].



(a) PCA (b) LDA (c) kPCA (d) SAe

Figure 5: Two-dimensional parameter-space representations using four data reduction techniques. The red data points are taken from parameter spaces described as bright and the blue points are described as *warm*. The solid black lines through the centroids mark dimensions of maximal variance in the data.

5. CONCLUSION

We have presented a model for the modulation of equalisation parameters using a two-dimensional control interface. The model utilises a stacked autoencoder to modify the dimensionality of the input data, and a weighting process that adapts the parameters to the LTAS of the input audio. We show that the autoencoder outperforms traditional models such as PCA and LDA in terms of reconstruction accuracy, and preserves timbral class discrimination in low-dimensional space. Whilst the model is limited by the inherent dimensionality of the data, we are able to achieve 92% reconstruction accuracy and 1.98 KLD separability on a dataset of 800 samples.

6. ACKNOWLEDGEMENTS

The work of the first author is supported by The Alexander S. Onassis Public Benefit Foundation.

7. REFERENCES

[1] R. Stables, S. Enderby, B. De Man, G. Fazekas, and J. D. Reiss, "SAFE: A system for the extraction and retrieval of semantic audio descriptors," in *ISMIR*. ISMIR, 2014.

[2] M. Sarkar, B. Vercoe, and Y. Yang, "Words that describe timbre: A study of auditory perception through language," in *Proceedings of the 2007 Language and Music as Cognitive Systems Conference*, 2007, pp. 11–13.

[3] J. W. Beauchamp, "Synthesis by spectral amplitude and "brightness" matching of analyzed musical instrument tones," *Journal of the Audio Engineering Society*, vol. 30, no. 6, pp. 396–406, 1982.

[4] E. Schubert and J. Wolfe, "Does timbral brightness scale with frequency and spectral centroid?," *Acta Acustica united with Acustica*, vol. 92, no. 5, pp. 820–825, 2006.

[5] J. Marozeau and A. de Cheveigné, "The effect of fundamental frequency on the brightness dimension of timbre," *Journal of the Acoustical Society of America*, vol. 121, no. 1, pp. 383–387, 2007.

[6] J. M. Grey, "Multidimensional perceptual scaling of musical timbres," *Journal of the Acoustical Society of America*, vol. 61, no. 5, pp. 1270–1277, 1977.

[7] A. Zacharakis, K. Pastiadis, J. D. Reiss, and G. Papadelis, "Analysis of musical timbre semantics through metric and non-metric data reduction techniques," in *Proceedings of the 12th International Conference on Music Perception and Cognition*, 2012, pp. 1177–1182.

[8] T. Brookes and D. Williams, "Perceptually-motivated audio morphing: Brightness," in *Proceedings of the 122nd Convention of the Audio Engineering Society*, 2007.

[9] A. Zacharakis and J. Reiss, "An additive synthesis technique for independent modification of the auditory perceptions of brightness and warmth," in *Proceedings of the 130th Convention of the Audio Engineering Society*, 2011.

[10] M. Cartwright and B. Pardo, "Social-EQ: Crowdsourcing an equalization descriptor map," in *ISMIR*, 2013, pp. 395–400.

[11] S. Mecklenburg and J. Loviscach, "subjEQt: Controlling an equalizer through subjective terms," in *CHI-06*, 2006, pp. 1109–1114.

[12] A. T. Sabin and B. Pardo, "2DEQ: an intuitive audio equalizer," in *Proceedings of the 7th ACM Conference on Creativity and Cognition*, 2009, pp. 435–436.

[13] T. Brookes and D. Williams, "Perceptually-motivated audio morphing: Warmth," in *Proceedings of the 128th Convention of the Audio Engineering Society*, 2010.

[14] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, no. 6, pp. 417, 1933.

[15] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[16] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[17] H. Drucker, C. J. C. Burges, , L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," *Advances in neural information processing systems*, vol. 9, pp. 155–161, 1997.

[18] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[19] E. Levina and P. J. Bickel, "Maximum likelihood estimation of intrinsic dimension," in *Advances in neural information processing systems*, 2004, pp. 777–784.