Cherla, S., Weyde, T., Garcez, A. d'Avila & Pearce, M. (2013). Learning Distributed Representations for Multiple-Viewpoint Melodic Prediction. Paper presented at the 14th International Society for Music Information Retrieval Conference, 4 - 8 Nov 2013, Curtiba, PR, Brazil.

CITY UNIVERSITY LONDON

A DISTRIBUTED MODEL FOR MULTIPLE-VIEWPOINT MELODIC PREDICTION

Srikanth Cherla[1,2], **Tillman Weyde**[1,2], **Artur d'Avila Garcez**[2] **and Marcus Pearce**[3]
[1]Music Informatics Research Group, Department of Computer Science, City University London
[2] Machine Learning Group, Department of Computer Science, City University London
[3]Centre for Digital Music, Queen Mary University of London
{srikanth.cherla.1, t.e.weyde, a.garcez}@city.ac.uk
marcus.pearce@eecs.qmul.ac.uk

ABSTRACT

The analysis of sequences is important for extracting information from music owing to its fundamentally temporal nature. In this paper, we present a distributed model based on the Restricted Boltzmann Machine (RBM) for melodic sequences. The model is similar to a previous successful neural network model for natural language [2]. It is first trained to predict the next pitch in a given pitch sequence, and then extended to also make use of information in sequences of note-durations in monophonic melodies on the same task. In doing so, we also propose an efficient way of representing this additional information that takes advantage of the RBM's structure. Results show that this RBM-based prediction model performs better than previously evaluated n-gram models in many cases. It is able to make use of information present in longer contexts more effectively than n-gram models, while scaling linearly in the number of free parameters required.

1. INTRODUCTION

Sequential structure in music influences our notions of musical style, similarity and the emotions we associate with it. The analysis of sequences in musical scores and equivalent symbolic representations of music is an integral part of Music Information Retrieval, with applications such as music classification [6], computational musicology [26], music creation [19], and music source separation [10]. In the past, this analysis has often been carried out using music generation systems [1, 4, 8, 13, 18].

The present research is based around previous work that adopted ideas proposed in information theory to music [7]. There, *Multiple-viewpoint Systems for Music Prediction* were introduced as a detailed re-interpretation of the key ideas of information theory [22] in music, through an analogy between language and musical style. In that work and what followed [21], Markov models were employed for learning melodic subsequences. While this is a reasonable choice, Markov models are often faced with a problem related to data sparsity known as the *curse of dimensionality*. This refers to the exponential rise in the number of model parameters with the length of the modelled subsequences. Recent research in Language Modelling has demonstrated that neural networks can be a suitable alternative to more widely used n-gram and variable-order Markov models [2, 5, 17]. There have been some initial results on the success of such models in music [3, 24].

In this paper, we present a model for melody prediction based on one such neural network — the Restricted Boltzmann Machine (RBM) [23]. The choice is motivated by the following. Firstly, the inherent non-linearity of the RBM makes it a suitable candidate for learning complex structures in data, such as those occurring in musical sequences. There exist efficient algorithms for training this model [11, 25]. The RBM, with its straightforward extensibility to deep networks [12], has become a vital building block for creating models that are capable of learning features from the data at multiple levels of abstraction.

We describe here a model for fixed-length subsequences of musical pitch, which compares favourably to n-gram models that were previously evaluated with a prediction task on a corpus of monophonic MIDI melodies [21]. This pitch-only version of the model is then adapted to also make use of note-durations in the melodies, on the same pitch-prediction task. In doing so, we also propose an efficient way to represent this additional information, which takes advantage of the RBM's structure and thus limits model complexity. The structure of the proposed model ensures that it scales only linearly with the length of subsequences to be learned and with the number of symbols in the data. We demonstrate an improvement of results by combining the two models in a manner similar to [7] using the arithmetic mean of their individual probability estimates. An implementation of the model in Python, along with scripts used to generate the results in this paper, are available upon request.

The remainder of this paper is organized as follows. The next section introduces music prediction and multiple viewpoint systems as a framework for music prediction. Section 3 explains the RBM and its discriminative interpretation which make up the basis for the model proposed

in this paper. This is followed by a description of the model itself in Section 4. An evaluation of the the model and its comparison with previously evaluated n-gram models is presented in Section 5, followed by discussion on possible directions for future research in Section 6.

2. MUSIC PREDICTION WITH MULTIPLE-VIEWPOINT SYSTEMS

In order to explain music prediction with multiple viewpoints, the analogy to natural language is used here. In statistical language modelling, the goal is to build a model that can estimate the joint probability distribution of subsequences of words occurring in a language L. A statistical language model (SLM) can be represented by the conditional probability of the next word w_T given all the previous ones $[w_1, \ldots, w_{(T-1)}]$ (written here as $w_1^{(T-1)}$), as

$$P(w_1^T) = \prod_{t=1}^{T} P(w_t | w_1^{(t-1)}) \; . \tag{1}$$

The most commonly used SLMs are *n-gram* models, which rely on the simplifying assumption that the probability of a word in a sequence depends only on the immediately preceding $(n-1)$ words [16]. This is known as the Markov assumption, and reduces (1) to

$$P(w_1^T) = \prod_{t=1}^{T} P(w_t | w_{(t-n+1)}^{(t-1)}) \; . \tag{2}$$

Following this approach, musical styles can be interpreted as vast and complex languages [7]. In music prediction, one is interested in learning the joint distribution of *musical event* sequences s_1^T in a *musical language* S. Much in the same way as an SLM, a system for music prediction models the conditional distribution $p(s_t | s_1^{(t-1)})$, or under the Markov assumption $p(s_t | s_{(t-n+1)}^{(t-1)})$. For each prediction, context information is obtained from the events $s_{(t-n+1)}^{(t-1)}$ immediately preceding s_t. Musical events have a rich internal structure and can be expressed in terms of directly observable or derived musical features such as pitch, note duration, inter-onset interval, or a combination of two or more such features. The framework of multiple-viewpoint systems for music prediction [7] was proposed in order to efficiently handle this rich internal structure of music by exploiting information contained in these different musical feature sequences, while at the same time limiting the dimensionality of the models using these features. In the interest of brevity, we limit ourselves to an informal discussion of multiple-viewpoint systems for monophonic music prediction and refer the reader to [7] for the underlying mathematical formulation.

A musical event s refers to the occurrence of a note in a melody. A *viewpoint type* (henceforth written as *type*) τ refers to any of a set of musical features that describe an event. The domain of a *type*, denoted by $|\tau|$ is the set of possible values of that type. A *basic type* is a directly observable or given feature such as *pitch, note duration, key-signature* or *time-signature*. A *derived type* can be derived from any of the basic types or other derived types. A *linked viewpoint type* is created by taking the Cartesian product over two or more types, thus "linking" them.

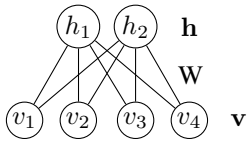

Figure 1. A simple Restricted Boltzmann Machine with four visible, two hidden, and no bias units.

A *multiple-viewpoint system* (MVS) is a set of models, each of which is trained on subsequences of one *type*, whose individual predictions are combined in some way to influence the prediction of the next event in a given event sequence. Given a context $s_{(t-n+1)}^{(t-1)}$ and an event s_t, each viewpoint τ in an MVS must compute the probability $p_\tau(s_t | s_{(t-n+1)}^{(t-1)})$. While originally n-gram models were proposed to be used with the multiple viewpoints framework, we demonstrate how a distributed model such as the RBM used here can serve as a scalable alternative.

3. RESTRICTED BOLTZMANN MACHINE

The Restricted Boltzmann Machine (RBM) is an undirected graphical model consisting of a set of r visible units $\mathbf{v}$ and a set of q hidden units $\mathbf{h}$. These make up the visible and hidden layers of the RBM respectively. The two layers are fully inter-connected but there exist no connections between any two hidden units, or any two visible units. In its original form, the RBM has binary, logistic units in both layers. Additionally, the units of each layer are connected to a bias unit whose value is always 1.

The edge between the i^{th} visible node and the j^{th} hidden node is associated with a weight w_{ji}. All these weights are together represented in a *weight matrix* $\mathbf{W}$ of size $q \times r$. The weights of connections between visible units and the bias unit are contained in an r-dimensional *visible bias* vector $\mathbf{b}$. Likewise, for the hidden units there is a q-dimensional *hidden bias* vector $\mathbf{c}$. The RBM is fully characterized by the parameters $\mathbf{W}$, $\mathbf{b}$ and $\mathbf{c}$. Figure 1 shows a simple RBM with four visible and two hidden units, without the bias unit to better illustrate its bipartite structure.

The activation probabilities of the units in the hidden layer given the visible layer (and vice versa) are given by the logistic sigmoid function as $p(h_j = 1|\mathbf{v}) = \sigma(c_j + W_{j\cdot}\mathbf{v})$, and $p(v_i = 1|\mathbf{h}) = \sigma(b_i + W'_{i\cdot}\mathbf{h})$ respectively. Due to the RBM's bipartite structure, the activation probabilities of the nodes within one of the layers are independent, if the activation of the other layer is given, i.e.

$$p(\mathbf{h}|\mathbf{v}) = \prod_{j=1}^{q} p(h_j|\mathbf{v}) \tag{3}$$

$$p(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^{r} p(v_i|\mathbf{h}) \; . \tag{4}$$

The RBM is a special case of the Boltzmann Machine, which is an energy-based model for representing probability distributions [15]. In such energy-based models, probability is expressed in terms of an energy function. In the case of the RBM, this function is expressed as

$$Energy(\mathbf{v}, \mathbf{h}) = -\mathbf{b}^\top \mathbf{v} - \mathbf{c}^\top \mathbf{h} - \mathbf{h}^\top \mathbf{W} \mathbf{v} \,. \quad (5)$$

Learning in energy-based models can be carried out in a *generative* fashion, by updating the weights and biases in order to minimize the overall energy of the system with respect to the training data. This amounts to maximizing the log-likelihood function of the joint probability distribution $p(\mathbf{v})$, which is given by

$$p(\mathbf{v}) = \frac{e^{-FreeEnergy(\mathbf{v})}}{Z}, \quad (6)$$

with $Z = \sum_{\mathbf{v}} e^{-FreeEnergy(\mathbf{v})}$, where

$$FreeEnergy(\mathbf{v}) = -\log \sum_{\mathbf{h}} e^{-Energy(\mathbf{v},\mathbf{h})} \,. \quad (7)$$

While computing the exact gradient of the log-likelihood function for $p(\mathbf{v})$ is not tractable, an approximation of this gradient called the Contrastive Divergence (CD) gradient has been found to be a successful update rule for training RBMs [11]. With the CD update, the RBM can be trained efficiently.

The RBM described above models the joint probability $p(\mathbf{v})$ of the set of visible units $\mathbf{v}$. However, as described in Section 2, we are interested in a conditional distribution of the form $p(y|\mathbf{x})$. It has been demonstrated in [14] how an RBM can be used for a *discriminative* task such as classification. The posterior class probability distribution of such an RBM has the form

$$p(\mathbf{y} = \mathbf{e}_c|\mathbf{x}) = \sum_{\mathbf{h}} p(\mathbf{y} = \mathbf{e}_c, \mathbf{h}|\mathbf{x}) \quad (8)$$

$$= \frac{e^{-FreeEnergy(\mathbf{x},\mathbf{e_c})}}{\sum_{c'=1\ldots C} e^{-FreeEnergy(\mathbf{x},\mathbf{e}_{\mathbf{c}'})}} \quad (9)$$

where $\mathbf{x}$ is the input vector, and $\mathbf{y}$ is a vector that is a *1-of-C* representation of the class (also known as *one-hot* encoding), with C being the number of classes. If $\mathbf{x}$ belongs to a class c, then $\mathbf{y} = \mathbf{e}_c$, where $\mathbf{e}_c$ is a vector with all values set to 0 except at position c. With respect to the RBM, $\mathbf{x}$ and $\mathbf{y}$ together make up the visible layer $\mathbf{v}$.

Assuming a training set $\mathcal{D}_{train} = \{(\mathbf{x}_i, y_i)\}$ where $\mathbf{x}_i$ and $y_i \in \{1, \ldots, C\}$ are the i-th input vector and target class respectively, training the RBM generatively involves minimizing the negative log-likelihood

$$\mathcal{L}_{gen}(\mathcal{D}_{train}) = -\sum_{i=1}^{|\mathcal{D}_{train}|} \log p(\mathbf{x}_i, \mathbf{y}_i) \,. \quad (10)$$

The RBM thus used in a discriminative manner, forms the basis of the prediction model described in the next section.

4. A DISTRIBUTED MODEL FOR USE WITH MULTIPLE VIEWPOINTS

The prediction model we present in this paper models the conditional distribution $p(s_t|s_{(t-n+1)}^{(t-1)})$. It places no restrictions on the *types* associated with events in the context $s_{(t-n+1)}^{(n-1)}$ (*input type*), or the predicted event s_t (*target type*). In the simplest case, both are the same. In the case where they are different, the performance of the model depends on how informative the input types are of the target type. In the present work, we demonstrate this model with two cases where (1) both the input and target viewpoint types are musical pitch, and (2) the input types are pitch and duration, and the target type pitch. The choice of the additional input type in the second case was motivated by simplicity and to lay emphasis on the representation.

For each monophonic melody (in MIDI format) in a given dataset, sequences of the relevant input and target types are first extracted using the MIDI Toolbox [9]. These values are encoded as binary 1-of-$|\tau|$ vectors, where $|\tau|$ is the size of the domain of type τ. In the case where more than one input type exists, their corresponding vectors are simply concatenated. Such an idea is similar to that of the linked viewpoint type proposed in [7]. There are however, two important distinctions between the two. Firstly, the input and target types must be identical in the case of the n-gram models originally proposed for use with multiple-viewpoint systems, whereas this is not a requirement for the RBM model. Secondly, a linked viewpoint between two arbitrary types τ_1 and τ_2 of domain sizes $|\tau_1|$ and $|\tau_2|$ respectively, would have a domain of size $|\tau_1| \times |\tau_2|$ in the case of the n-gram models. Thus, for subsequences of length n, the number of free parameters to be estimated are $(|\tau_1| \times |\tau_2|)^n$ in the worst case. In contrast, the number to be estimated in case of the RBM model, with q hidden units and r visible units, is $(q \times r) + q + r$, where $r = (n-1) \times [(|\tau_1|+1)+(|\tau_2|+1)]+|\tau_3|$, and τ_3 the target type. The additional visible unit added to those of each of the types in the context (τ_1 and τ_2) is 1 when the corresponding context event is absent at the start of a melody. Such a model only scales linearly with the length of the learned subsequences as well as the domain size of each of the involved viewpoint types (assuming q is constant). Its structure is depicted in Figure 2. Here we considered only those cases with a single target type.

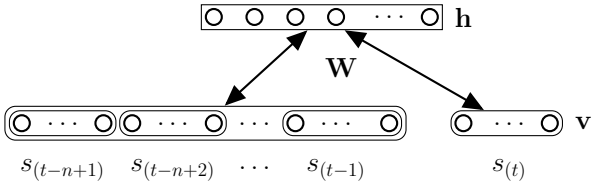

Figure 2. The structure of the prediction model. The set of nodes in the visible layer grouped together on the left make up the context $s_{(t-n+1)}^{(t-1)}$ of the input type(s). The set of nodes $s_{(t)}$ to the far right corresponds to the target type.

To train the model generatively, a subsequence $s_{(t-n+1)}^{t}$ is clamped to all the nodes in the visible layer. Training is done using the first instantiation of the Contrastive Divergence learning algorithm (CD-1). This simply means that the model parameters are updated after a single step of Gibbs sampling [11]. During prediction, the probability of each of the possible pitches in the prediction space is determined using (9). The distribution generated in this way does not require any kind of smoothing operation for unseen subsequences unlike n-gram models, where in [21] an empirical evaluation of different smoothing techniques was found necessary to establish the most reliable one.

5. EVALUATION

In order to evaluate the proposed prediction model, we make a comparison to a previous study of n-gram models for music prediction in [21]. There, *cross-entropy* was used to measure the information content of the models. This quantity is related to *entropy*, which is defined as

$$H(p) = -\sum_{s \in S} p(s) \log_2 p(s) \; . \tag{11}$$

where $p(s \in S) = p(\chi = s)$ is the probability mass function of a random variable χ distributed over a discrete alphabet $S = \{s_1, \ldots, s_k\}$ such that the individual probabilities are independent and sum to 1. The value of entropy, with reference to a prediction model, is a measure of the uncertainty of its predictions. A higher value reflects greater uncertainty. In practice, one rarely knows the true probability distribution of the stochastic process and uses a model to approximate the probabilities in (11). An estimate of the goodness of this approximation can be measured using cross-entropy (H_c) which represents the divergence between the entropy calculated from the estimated probabilities and the source model. This quantity can be computed over all the subsequences of length n in the test data $\mathcal{D}_{test}$, as

$$H_c(p_{mod}, \mathcal{D}_{test}) = \frac{-\sum_{s_1^n \in \mathcal{D}_{test}} \log_2 p_{mod}(s_n | s_1^{(n-1)})}{|\mathcal{D}_{test}|} \tag{12}$$

where p_{mod} is the probability assigned by the model to the last pitch in the subsequence given its preceding context. Cross-entropy approaches the true entropy as the number of test samples ($|\mathcal{D}_{test}|$) increases.

Evaluation was carried out on a corpus of monophonic MIDI melodies that cover a range of musical styles. The corpus is a collection of 8 datasets containing a total of $54,308$ musical events and was also used to evaluate n-gram models for music prediction in [21]. There, two different models were evaluated both individually and in combination. The first of these was a Long-Term Model (LTM), that was governed by structure and statistics induced from a large corpus of sequences from the same musical style. And the other was a Short-Term Model (STM) which relied on structure and statistics particular to the melody being predicted. The prediction model presented here deals only with long-term effects that are induced from a corpus, and is thus compared with the two best performing LTMs in [21] of unbounded order (labelled there as *C*I*) and order bound 2 respectively. To facilitate a direct comparison between the two approaches, the melodies are not transposed to a default key.

For the RBM model, different hyperparameters were evaluated through a grid search over the learning rate $\lambda = \{0.01, 0.05\}$, the number of hidden units $n_{hid} = \{100, 200, 400\}$, and the weight-cost $w_{cost} = \{0.0001, 0.0005\}$. Each model was trained using mini-batch gradient descent over 500 epochs with a batch size of 100 samples. The momentum μ, was set to 0.5 during the first five epochs and then increased to 0.9 for the rest of the training. Each model was evaluated with 10-fold cross-validation.

We carry out three types of evaluation. The first measures the information content of the pitch-only version of the proposed model using cross-entropy, and compares it to the n-gram models of [21]. It was observed that the RBM model compares favourably with the best of the n-gram models by making better use of information in longer contexts. In the second evaluation, we compare a variant of the model with input types pitch and duration and target type pitch to its pitch-only counterpart. And lastly, we combine these two models using mixture-of-experts and demonstrate how this can further improve the model performance in comparison to the individual models.

The first evaluation is carried out with cross-validation separately for each of the individual datasets. The context length is varied between 1 and 8. It was found that the RBM models with context length greater than 2 perform better than corresponding n-gram models on average. This is illustrated in Figure 3. An RBM model of suitable context length perform marginally better than the best-performing n-gram model — that of unbounded order. The same is the case with the best bounded-order n-gram model (of context length 2) and the RBM model of the same context length. While it was found that the performance of bounded order n-gram models tends to worsen on further increasing the context length, the performance of RBM models continues to improve until a context length of 4. The value of n where the RBM model performs better than the n-gram models of unbounded order is different on different datasets, and typically occurs between $n = 3$ and $n = 7$. The best average model cross-entropy of 2.819 is reached for a context length of 4. For models using longer contexts an increase in training performance was accompanied by a slight worsening of test performance, indicating overfitting. We suspect that the overall performance of the RBM models can be further improved with an optimized grid-search strategy in the hyper-parameter space, but leave this to be explored in the future. The optimal number of hidden units in our search was 100 across all datasets for almost all context lengths, leading to a linear increase in model size with context length.

In the second evaluation, we compared the cross-entropies of the single and multiple input type models (pitch and pitch with duration respectively) using the same target type

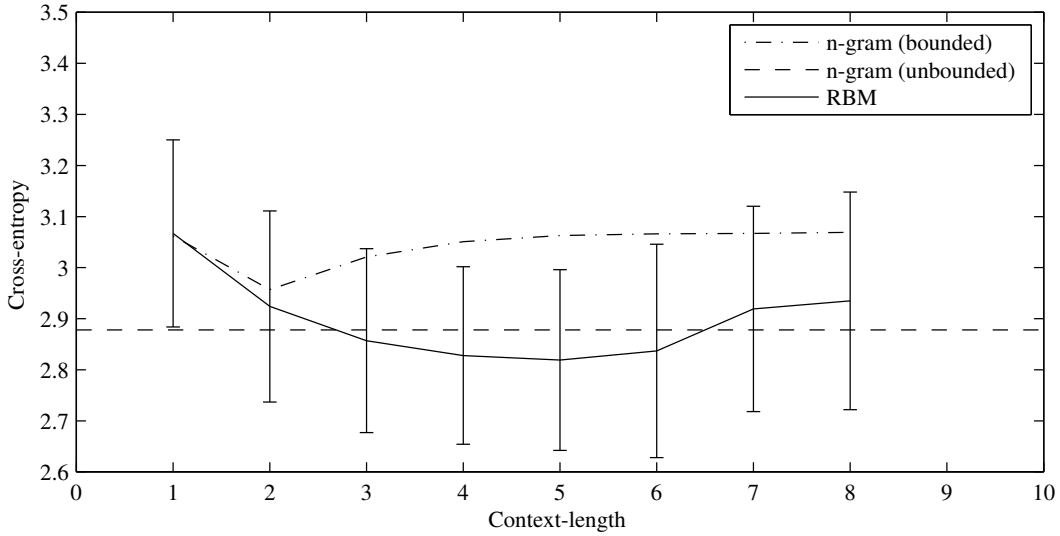

Figure 3. Variation in average cross-entropy of the prediction models with context length l (with standard deviation across folds for the RBM model). The cross-entropy of the RBM models progressively decreases until $l = 4$, while that of the n-gram models evaluated in [21] is minimal at $l = 2$ and increases thereafter. The performance of the n-gram model of unbounded order is indicated by the dashed line.

(pitch), on the Bach chorale subset of the corpus. The results are shown in Table 1. The choice of adding duration was motivated by simplicity but the results show that it was not ideal for improving predictions. This conclusion is also supported by a similar trend observed with n-gram models, where a small deterioration in performance was observed on adding duration. The RBM model shows small performance improvements for some context lengths. This indicates that the representation for multiple input types proposed in Section 4 as an alternative to the linked viewpoints may indeed be effective.

l	1	2	3	4
n-gram (p)	2.737	2.565	2.505	2.473
n-gram (p + d)	2.761	2.562	2.522	2.502
RBM (p)	2.698	2.530	2.490	2.470
RBM(p + d)	2.660	2.512	2.481	2.519
RBM (combined)	2.663	2.486	2.462	2.413

Table 1. Cross-entropies of the single (pitch) and multiple (pitch, duration) input type RBM models and their combination over a range of context lengths l on the Bach chorales dataset. The individual RBM models compare favourably with corresponding n-gram models.

To illustrate the application of the proposed RBM model to multiple viewpoints for music prediction, we combine the pitch-only and the pitch & duration models. We use a simple mixture-of-experts model, i.e., take the arithmetic mean of the distributions each of the two models predicts for pitch. The results of this are listed in the third row of Table 1 and show an improvement over individual models.

6. CONCLUSIONS & FUTURE WORK

We presented a distributed model based on the Restricted Boltzmann Machine for multiple-viewpoint music prediction. It was demonstrated how such a model can be a scalable alternative to n-gram models for simultaneously modelling sequences of multiple musical features. The proposed model was evaluated in comparison with n-gram models and was found to compare favourably with them. It is able to make better use of information in longer event contexts than n-gram models, and also scales linearly with context length.

In the future, we would first like to address some of the issues left open in the present research. These include experiments with more promising viewpoint-type combinations as reported in [7] and [20], the use of alternative data fusion techniques like the weighted mixture- and product-of-experts [20], and further optimization of the existing model parameters. Previous research suggests that combining the LTM and STM improves prediction performance [7, 20] and, in fact, the combined n-gram model reported in [20] (mean cross-entropy: 2.479 for all datasets; 2.342 for the chorale dataset) outperforms the long-term RBMs examined here. Given the improved performance of these long-term RBMs, we expect adding a short-term component will yield the best prediction performance yet observed for this corpus. Extensions of the present model to handle polyphony and higher-level musical structure will also be explored. We would also like to apply the prediction model described here to some of the MIR tasks listed in Section 1. The present model can be potentially extended into a deep network, as demonstrated in [11], which is expected to improve its performance further.

7. ACKNOWLEDGEMENTS

Srikanth Cherla is supported by a Ph.D. studentship from City University London. The authors would like to thank Son Tran for many useful discussions on RBMs, and the reviewers for their valuable feedback on the paper.

8. REFERENCES

[1] Charles Ames. The Markov Process as a Compositional Model: A Survey and Tutorial. *Leonardo*, 22(2):175–187, 1989.

[2] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.

[3] Greg Bickerman, Sam Bosley, Peter Swire, and Robert Keller. Learning to Create Jazz Melodies using Deep Belief Nets. In *International Conference On Computational Creativity*, 2010.

[4] John Biles. Genjam: A genetic algorithm for generating jazz solos. In *Proceedings of the International Computer Music Conference*, pages 131–131, 1994.

[5] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

[6] Darrell Conklin. Multiple viewpoint systems for music classification. *Journal of New Music Research*, 42(1):19–26, 2013.

[7] Darrell Conklin and Ian H Witten. Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24(1):51–73, 1995.

[8] David Cope. *Experiments in musical intelligence*, volume 12. AR Editions Madison, WI, 1996.

[9] Tuomas Eerola and Petri Toiviainen. MIR in Matlab: The Midi Toolbox. In *Proceedings of the International Conference on Music Information Retrieval*, pages 22–27. Universitat Pompeu Fabra Barcelona, 2004.

[10] Joachim Ganseman, Paul Scheunders, Gautham J Mysore, and Jonathan S Abel. Evaluation of a Score-informed Source Separation System. In *International Society for Music Information Retrieval Conference (ISMIR)*, pages 219–224, 2010.

[11] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[12] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18:1527–1554, 2006.

[13] Robert M Keller and David R Morrison. A Grammatical Approach to Automatic Improvisation. In *Sound and Music Computing Conference*, pages 11–13, 2007.

[14] Hugo Larochelle and Yoshua Bengio. Classification using discriminative restricted Boltzmann machines. In *International Conference on Machine Learning (ICML)*, pages 536–543. ACM Press, 2008.

[15] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting Structured Data*, 2006.

[16] Christopher D Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

[17] Andriy Mnih and Geoffrey E Hinton. A scalable hierarchical distributed language model. In *Advances in neural information processing systems*, pages 1081–1088, 2008.

[18] Michael C Mozer. Connectionist music composition based on melodic, stylistic and psychophysical constraints. *Music and connectionism*, pages 195–211, 1991.

[19] Francois Pachet. The continuator: Musical interaction with style. *Journal of New Music Research*, 32(3):333–341, 2003.

[20] Marcus Pearce. *The Construction and Evaluation of Statistical Models of Melodic Structure in Music Perception and Composition*. PhD thesis, 2005.

[21] Marcus Pearce and Geraint Wiggins. Improved methods for statistical modelling of monophonic music. *Journal of New Music Research*, 33(4):367–385, 2004.

[22] Claude E. Shannon. A Mathematical Theory of Communication. *The Bell System Techincal Journal*, 27(July):379–423, 623–656, 1948. Reprinted in *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.

[23] Paul Smolensky. Parallel distributed processing: explorations in the microstructure of cognition, vol. 1. chapter Information processing in dynamical systems: foundations of harmony theory, pages 194–281. MIT Press, Cambridge, MA, USA, 1986.

[24] Athina Spiliopoulou and Amos Storkey. Comparing probabilistic models for melodic sequences. In *Machine Learning and Knowledge Discovery in Databases*, pages 289–304. 2011.

[25] Tijmen Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pages 1064–1071. ACM, 2008.

[26] Raymond Whorley, Christophe Rhodes, Geraint Wiggins, and Marcus Pearce. Harmonising melodies: Why do we add the bass line first? In *International Conference on Computational Creativity*, pages 79–86, 2013.