RENDERING EXPRESSIVE PERFORMANCES OF MUSICAL PIECES THROUGH SAMPLING FROM GENERATIVE PROBABILISTIC MODELS

CARLOS EDUARDO CANCINO CHACÓN AND MAARTEN GRACHTEN

Austrian Research Institute for Artificial Intelligence

OFAI-TR-2014-01 Version 1.0

Abstract. The Basis Modeling (BM) framework is a state-of-the-art model for musical expression that has been used in both analysis and rendering of expressive music performances. In their current form, these models are deterministic, and thus, given a trained model, there is only one possible performance that can be generated for a given piece. By using a Bayesian framework, it is possible to produce a probabilistic interpretation of the models, and then generate performances by sampling from their predictive distributions. In this report we provide detailed derivations of the predictive distributions both the linear and non-linear versions of the BM approach.

Musical expression, probabilistic basis modeling, Bayesian non-linear regression

1. Introduction

Computational models of musical expression can serve analytical purposes, to test hypotheses and gain knowledge about the complex human skill of music performance, but they can also be used as a tool to produce expressive renderings of musical scores. Non-probabilistic models typically produce such renderings by a deterministic procedure, such that a musical score gives rise to only a single performance. Probabilistic models on the other hand, lend themselves to sampling, producing multiple interpretations that follow a (possibly multi-modal) distribution, estimated from recorded performances.

An existing computational modeling framework for music expression, the Basis Modeling (BM) framework proposed by Grachten and Widmer [2012], has been shown successful as a modeling approach for classical piano performance. Subsequent, non-linear versions of the model improved the original linear model in terms of modeling accuracy [Cancino Chacón and Grachten, 2015; Grachten and Cancino Chacón, 2016], but the models did not involve any probabilistic modeling, and could not be used to produce distinct performances for a given musical piece.

In this document, we describe probabilistic interpretations of both the linear and the non-linear versions of the BM framework. Based on this reformulation, we derive the *predictive distributions* for each of the models, that is, the distributions over the random variables for which predictions are made. In the BM framework these variables (also called *expressive parameters*), determine what a performance will sound like in terms of dynamics, tempo, timing, and articulation. Expressive performances can be generated by sampling from those distributions.

The rest of this document is structured as follows. In Section 2 we provide a brief overview of the BM framework in its linear and non-linear versions. In Section 3, a probabilistic interpretation of the models using a Bayesian framework under the assumption of Gaussian priors is given. Detailed derivations of the predictive distributions for both the linear and non-linear versions of the BM approach are provided in Section 4. Finally, Conclusions and future work is provided in Section 5

2. Basis Models for Musical Expression

The BM approach can be understood as the use of *basis-functions* $\boldsymbol{\varphi}$, numerical descriptors that encode certain aspects of a musical score $\mathcal{X}$, to model a set of expressive parameters $\mathbf{t}$[1], quantitative descriptors of which characterize aspects of musical expression. In this paper, we consider a *musical score* as a sequence of elements that hold musical information, such as note elements (e.g. pitch, duration) and non-note elements (e.g. *piano*, *forte*, *crescendo*) [Grachten and Widmer, 2012]. Formally, if we denote the set of all note elements in a score by $\mathcal{X}$, a basis function is a real valued mapping in the form $\varphi \colon \mathcal{X} \mapsto \mathbb{R}$. Figure 1 schematically illustrates the modeling of an expressive parameter, in this case expressive dynamics, using basis-functions.

Let $\mathbf{x} = (x_1, \dots, x_N)^T \in \mathbb{R}^N$ be a vector representing a set of N notes in a musical score, $\boldsymbol{\varphi}(x_i) = (\varphi_1(x_i), \dots, \varphi_M(x_i))^T \in \mathbb{R}^M$ be a vector whose elements are the values of the basis functions for note x_i, and $\boldsymbol{\Phi} \in \mathbb{R}^{N\times M}$ is a matrix with elements $\Phi_{ij} = \varphi_j(x_i)$ representing the whole musical score. The numerical values of the expressive parameters for each note predicted by a BM model, i.e. the output of the model, are represented by $\mathbf{y} = (y_1, \dots, y_N)^T \in \mathbb{R}^N$. Furthermore, we can join all expressive parameters into a matrix $\mathbf{Y} \in \mathbb{R}^{N\times K}$ such that each of its columns corresponds to a vector of expressive parameters. In this way, we can model the expressive parameters as a function of the input basis functions as

$$\mathbf{Y} = f(\boldsymbol{\Phi}; \mathbf{w}), \tag{1}$$

where $f(\cdot)$ is a function of $\boldsymbol{\varphi}$ given parameters $\mathbf{w}$. In the following several choices for $f(\cdot)$ will be described.

2.1. Linear Basis Models (LBMs).

The simplest way to explore the influence of the basis functions in the expressive parameters is using a linear regression. In this way, we can model each expressive parameters $\mathbf{y}_i$ as weighted sum of the basis functions, i.e.

$$\mathbf{y}_i = f_{lbm}(\boldsymbol{\Phi}; \mathbf{w}^{(i)}) = \boldsymbol{\Phi}\mathbf{w}^{(i)}, \tag{2}$$

[1] We use the terms *expressive parameters*, *expressive targets*, or simply *targets* interchangeably to refer to $\mathbf{t}$

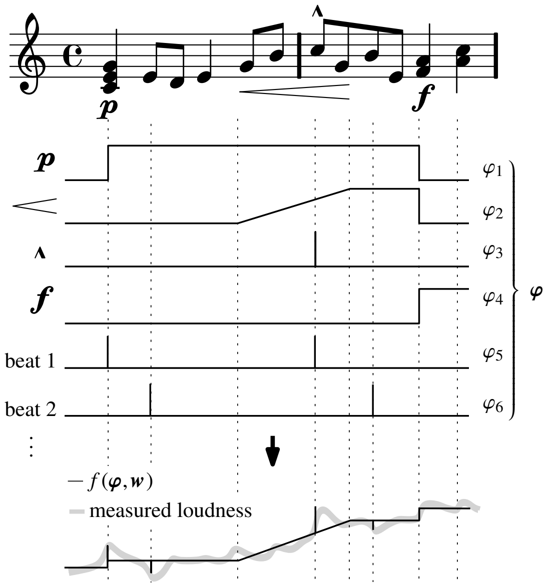

FIGURE 1. Schematic view of expressive dynamics as a function $f(\boldsymbol{\varphi}; \boldsymbol{w})$ of basis-functions $\boldsymbol{\varphi} = (\varphi_1, \cdots, \varphi_6)$, representing dynamic annotations and metrical basis functions.

where, $\mathbf{w}_i \in \mathbb{R}^M$ is a vector of weights. This is equivalent as writing

$$\mathbf{Y} = f_{lbm}(\mathbf{\Phi}; \mathbf{w}) = \mathbf{\Phi}\mathbf{w}, \tag{3}$$

where, $\mathbf{w} = [\mathbf{w}^{(i)} \ \dots \ \mathbf{w}^{(K)}] \in \mathbb{R}^{M \times K}$ is a matrix of weights. For a more detailed description of this model see [Grachten and Widmer, 2012; Grachten et al., 2014].

2.2. **Non-linear Basis Models (NBMs).** The influence of the basis functions in the expressive parameter can be modeled in a non-linear way using Feed Forward Neural Networks (FFNNs). These neural networks can be described as a series of (non-linear) transformations of the input data [Bishop, 2006]. In this way, we can write the expressive parameters as the output of a fully-connected FFNN with L *hidden layers* as

$$\mathbf{Y} = f_{nbm}(\mathbf{\Phi}; \mathbf{w}) = f^{(L)}\left(\mathbf{H}^{(L-1)}\mathbf{w}^{(L)} + \mathbf{w}_0^{(L)}\right) \tag{4}$$

where $\mathbf{H}^{(l)} \in \mathbb{R}^{N \times D_l}$ is the activation of the l-th hidden layer for all N timesteps, given by

$$\mathbf{H}^{(l)} = f^{(l-1)}\left(\mathbf{H}^{(l-1)}\mathbf{w}^{(l)} + \mathbf{w}_0^{(l)}\right) \tag{5}$$

and $\mathbf{H}^{(0)} = \mathbf{\Phi}$. The set of all parameters is $\mathbf{w} = \{\mathbf{w}_0^{(1)}, \mathbf{w}^{(1)}, \dots, \mathbf{w}_0^{(L)}, \mathbf{w}^{(L)}\}$, where $\mathbf{w}^{(l)}$ are the weights and $\mathbf{w}_0^{(l)}$ are the biases of the l-th hidden layer. The activation function of the l-th layer is represented by $f^{(l)}$. Common (non-linear) activation functions are sigmoid, hyperbolic tangent, softmax and rectifier ($ReLU(x) = \max(0, x)$). For a more technical description of the NLBM, we refer the reader to [Cancino Chacón and Grachten, 2015].

2.3. **Recurrent Non-linear Basis Models (RNBMS).** Both the LBM and NBM models are static model that do not allow for modeling temporal dependencies within parameters. This problem can be addressed by using Recurrent Neural Networks (RNNs). The basic structure of an RNN is the recurrent layer. The output of one such layer at time t, i.e. the t-th row of the recurrent layer $\mathbf{H}^{(r)}$, can be written as

$$\mathbf{h}_t^{\mathrm{T}} = f_h \left(g_\varphi \left(\boldsymbol{\varphi}(x_t) \right) + g_h \left(\mathbf{h}_{t^*} \right) \right)^{\mathrm{T}}, \tag{6}$$

where $g_\varphi \left(\boldsymbol{\varphi}(x_t) \right)$ represents the contribution of the input of the network at time t, $g_h^{(l)} \left(\mathbf{h}_{t^*} \right)$ is the contribution of other time steps (past or future, or a combination of both) of the state of the recurrent layer. As in the case of NBMs, $f_h(\cdot)$ is an element-wise (non-linear) activation function. The output of the network can be computed in a similar fashion to traditional FFNNs using Equation (4), where the non-recurrent and recurrent hidden layers are computed using Equations (5) and (6), respectively. A more mathematical formulation of RNNs can be found in [Graves, 2013] For a more detailed description of the RNBMs see [Grachten and Cancino Chacón, 2016].

3. Probabilistic interpretation of the Basis Models

We can use a Bayesian framework, to provide a probabilistic interpretation of the BM framework. In order to differentiate from the deterministic outputs of the models above, for each expressive parameter we introduce a vector of expressive targets $\mathbf{t} \in \mathbb{R}^N$. Each of these vectors is a random variable, related to their equivalent expressive parameters as

$$\mathbf{t} = \mathbf{y} + \boldsymbol{\epsilon} \tag{7}$$

where $\boldsymbol{\epsilon}$ is a zero mean Gaussian noise with precision β^{-1}. These targets can be similarly concatenated into a matrix as $\mathbf{T} \in \mathbb{R}^{N \times K}$. In a similar fashion, and with a slight abuse of notation, we can write the matrix of expressive targets as

$$\mathbf{T} = \mathbf{Y} + \boldsymbol{\epsilon} = f(\boldsymbol{\Phi}, \mathbf{w}) + \boldsymbol{\epsilon}, \tag{8}$$

where $\boldsymbol{\epsilon}$ is a zero mean Gaussian noise of the same dimensions as $\mathbf{Y}$. A set of O musical scores is denoted as

$$\boldsymbol{\mathcal{X}} = \{\boldsymbol{\Phi}_1, \dots, \boldsymbol{\Phi}_O\}, \tag{9}$$

where $\boldsymbol{\Phi}_i$ is the basis–functions–encoding of score $\mathcal{X}_i$, with their corresponding set of expressive targets being

$$\boldsymbol{\mathcal{T}} = \{\mathbf{T}_1, \dots, \mathbf{T}_O\}. \tag{10}$$

We make the simplifying assumption that the matrix of expressive targets $\mathbf{T}_i$ corresponding to score $\mathcal{X}_i$ is independent and identically distributed (iid) to all other matrices of expressive targets in $\boldsymbol{\mathcal{T}}$. In musical terms, this assumption is equivalent to consider an expressive performance of score $\mathcal{X}_i \in \boldsymbol{\mathcal{X}}$ to be independent to the performance of score $\mathcal{X}_j \in \boldsymbol{\mathcal{X}}$ for all $i \neq j$. This assumption does not hold in situations where the performance of a piece is influenced by the performance of previous and/or future pieces, as would be the case of a live performance.

From Equation (8), it follows that the conditional distribution of $\mathbf{T}$ given the parameters $\mathbf{w}$ is given by

$$p(\mathbf{T} \mid \mathbf{w}) = \mathcal{N}(\mathbf{T} \mid f(\mathbf{\Phi}; \mathbf{w}), \beta^{-1}). \tag{11}$$

Since we assumed that the targets are iid, the conditional distribution of the set of targets $\mathcal{T}$ is given by

$$p(\mathcal{T} \mid \mathbf{w}) = \prod_{o=1}^{O} p(\mathbf{T}_o \mid \mathbf{w}). \tag{12}$$

The joint probability distribution of the targets $\mathcal{T}$ and the parameters $\mathbf{w}$ is given by

$$p(\mathcal{T}, \mathbf{w}) = p(\mathcal{T} \mid \mathbf{w})p(\mathbf{w}), \tag{13}$$

where $p(\mathbf{w})$ is the prior distribution of the parameters. We assume this prior probability distribution to be Gaussian, i.e.

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w} \mid \mathbf{m}_0, \mathbf{S}_0), \tag{14}$$

where $\mathbf{m}_0$ is the prior mean and $\mathbf{S}_0$ is the prior covariance matrix.

Using Bayes' Theorem, the posterior probability distribution of $\mathbf{w}$ given $\mathcal{T}$ is given by

$$p(\mathbf{w} \mid \mathcal{T}) = \frac{p(\mathcal{T} \mid \mathbf{w})p(\mathbf{w})}{\int p(\mathcal{T} \mid \mathbf{w})p(\mathbf{w})d\mathbf{w}} \tag{15}$$

For estimation problems, it is useful to rewrite the above posterior distribution in the log domain as

$$\log p(\mathbf{w} \mid \mathcal{T}) = \log p(\mathcal{T} \mid \mathbf{w}) + \log p(\mathbf{w}) - \underbrace{\log \int_{\mathbf{w}} p(\mathcal{T} \mid \mathbf{w})p(\mathbf{w})d\mathbf{w}.}_{\text{does not depend on } \mathbf{w}} \tag{16}$$

The maximum-a-posteriori (MAP) estimation of the parameters involves maximizing the log posterior probability, i.e.,

$$\begin{aligned} \mathbf{w}_{map} &= \operatorname*{argmax}_{\mathbf{w}} \left(\log p(\mathbf{w} \mid \mathcal{T})\right) \\ &= \operatorname*{argmax}_{\mathbf{w}} \left(\underbrace{\log p(\mathcal{T} \mid \mathbf{w}) + \log p(\mathbf{w})}_{\mathcal{L}_{map}(\mathbf{w})} \right). \end{aligned} \tag{17}$$

3.1. **LBMs.** For the linear models, the MAP solution can be given analytically, since the posterior distribution is Gaussian. For each expressive target $\mathbf{t}_i$, the solution that maximizes the cost function is given by

$$\mathbf{w}^{(i)}_{map;lbm} = \left(\beta \begin{bmatrix} \mathbf{\Phi}_1 \\ \vdots \\ \mathbf{\Phi}_O \end{bmatrix}^T \begin{bmatrix} \mathbf{\Phi}_1 \\ \vdots \\ \mathbf{\Phi}_O \end{bmatrix} + \mathbf{S}_0^{-1} \right)^{-1} \left(\beta \begin{bmatrix} \mathbf{\Phi}_1 \\ \vdots \\ \mathbf{\Phi}_O \end{bmatrix}^T \begin{bmatrix} \mathbf{t}_{1,i} \\ \vdots \\ \mathbf{t}_{O,i} \end{bmatrix} + \mathbf{S}_0^{-1}\mathbf{m}_0 \right). \tag{18}$$

The prior parameters $\mathbf{m}_0$ and $\mathbf{S}_0$ can be computed using the expectation-maximization (EM) algorithm [Bishop, 2006]. A detailed derivation of these parameters, as well as further details of the general MAP solution for LBMs can be found in [Cancino Chacon et al., 2014].

3.2. **(R)NBMs.** The following derivation applies to both the NBM and RNBM models. Given the non-linear dependencies of the parameters and the output the (R)NBM models, the posterior distribution given by Equation (15) is no longer Gaussian. Since $\mathbf{T}$ is a matrix, the resulting conditional distribution is a matrix Gaussian distribution. We assume that $\mathrm{vec}(\mathbf{T}) \sim \mathcal{N}\left(\mathrm{vec}(f(\mathbf{\Phi}, \mathbf{w})), \beta^{-1}\mathbf{I}\right)$. We can simplify the prior distribution of the parameters over the weights given by Equation (14) by choosing $\mathbf{m}_0 = \mathbf{0}$ and $\mathbf{S}_0 = \alpha^{-1}\mathbf{I}$. Substituting in Equation (15), we can write the log posterior distribution as

$$\log p(\mathbf{w} \mid \boldsymbol{\mathcal{T}}) = -\frac{\beta}{2}\sum_{o=1}^{O} \|f(\mathbf{\Phi}_o, \mathbf{w}) - \mathbf{T}_o\|^2 - \frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \text{const.} \tag{19}$$

It is straightforward to see that maximizing the posterior distribution is equivalent to minimizing a the regularized squared error, i.e.

$$\mathbf{w}_{map;(r)nbm} = \underset{\mathbf{w}}{\operatorname{argmin}} \left(\frac{\beta}{2}\sum_{o=1}^{O} \|f(\mathbf{\Phi}_o, \mathbf{w}) - \mathbf{T}_o\|^2 + \frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} \right). \tag{20}$$

This minimization can be accomplished using an algorithm of the family of stochastic gradient descent (SGD) optimization methods, such as RMSProp [Dauphin et al., 2015], with the gradient of the loss function being computed through backpropagation for NBMs and backpropagation through time for RNBMs [Rumelhart et al., 1986]. The precision parameters α and β can be computed using the EM algorithm [Bishop, 2006].

4. Predictive Distribution

In previous sections we developed probabilistic extensions of the different versions of the BM approach. Given a model trained over a set of scores $\boldsymbol{\mathcal{X}}$ and its respective set of expressive targets $\boldsymbol{\mathcal{T}}$, we can make predictions $\mathbf{T}$ for a new score $\mathbf{\Phi}$. These predictions can be expressed in terms of the predictive distribution over $\mathbf{T}$, rather than simply a point estimate. The predictive distribution is obtained by marginalizing with respect to the posterior distribution of the parameters, i.e.

$$p(\mathbf{T} \mid \boldsymbol{\mathcal{T}}) = \int p(\mathbf{T} \mid \mathbf{w}) p(\mathbf{w} \mid \boldsymbol{\mathcal{T}}) d\mathbf{w}, \tag{21}$$

where the conditional probability $p(\mathbf{T} \mid \mathbf{w})$ is given as in Equation (11), and $p(\mathbf{w} \mid \boldsymbol{\mathcal{T}})$ is given as in Equation (15).

4.1. **LBMs.** We can write the predictive distribution for the i-th expressive target as follows. Due to the Gaussian assumptions on the priors and the linear dependency of the model parameters $\mathbf{w}$, the posterior probability $p(\mathbf{w} \mid \boldsymbol{\mathcal{T}})$ is also Gaussian given by[2]

$$p(\mathbf{w}^{(i)} \mid \boldsymbol{\mathcal{T}}) = \mathcal{N}(\mathbf{w}^{(i)} \mid \mathbf{m}_N, \mathbf{S}_N), \tag{22}$$

where $\mathbf{m}_N$ is the posterior mean and $\mathbf{S}_N$ is the posterior covariance matrix, calculated as

$$\mathbf{m}_N = \mathbf{S}_N \left(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta \begin{bmatrix} \boldsymbol{\Phi}_1 \\ \vdots \\ \boldsymbol{\Phi}_O \end{bmatrix}^T \begin{bmatrix} \mathbf{t}_{1,i} \\ \vdots \\ \mathbf{t}_{O,i} \end{bmatrix} \right) \quad \text{and} \quad \mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \beta \begin{bmatrix} \boldsymbol{\Phi}_1 \\ \vdots \\ \boldsymbol{\Phi}_O \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\Phi}_1 \\ \vdots \\ \boldsymbol{\Phi}_O \end{bmatrix}. \tag{23}$$

Substituting Equations (2), (11) and (22) in Equation (21) results in the marginalization of a Gaussian distribution, which is also Gaussian[3]. Therefore, we can write the predictive distribution for the i-th expressive target as

$$p(\mathbf{t}_i \mid \boldsymbol{\mathcal{T}}) = \mathcal{N}(\mathbf{t}_i \mid f_{lbm}(\boldsymbol{\Phi}, \mathbf{w}^{(i)}_{map;lbm}), \sigma_N^2), \tag{24}$$

where

$$\sigma_N^2 = \frac{1}{\beta}\mathbf{I} + \boldsymbol{\Phi}^{\mathrm{T}} \left(\mathbf{S}_0^{-1} + \beta \begin{bmatrix} \boldsymbol{\Phi}_1 \\ \vdots \\ \boldsymbol{\Phi}_O \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \boldsymbol{\Phi}_1 \\ \vdots \\ \boldsymbol{\Phi}_O \end{bmatrix} \right)^{-1} \boldsymbol{\Phi}. \tag{25}$$

4.2. **(R)NBMs.** For the both non-linear models, the integral in Equation (21) is analytically intractable, due to the posterior probability of the parameters being non-Gaussian, and due to the non-linear dependencies of the parameters in the conditional distribution of the target variables. In order to solve this problem, we can take two alternatives:

(1) *Linear-Gaussian approximation.* We can use the Laplace approximation [Bishop, 1995] to find a Gaussian approximation to the posterior distribution as

$$p(\mathbf{w} \mid \boldsymbol{\mathcal{T}}) \approx q(\mathbf{w} \mid \boldsymbol{\mathcal{T}}) = \mathcal{N}\left(\mathbf{w} \mid \mathbf{w}_{map;(r)nbm}, \mathbf{A}^{-1}\right), \tag{26}$$

where $\mathbf{A}$ is the matrix of second derivatives of the negative log posterior distribution. From Equation (19), this is given by

$$\mathbf{A} = -\nabla_{\mathbf{w}}\nabla_{\mathbf{w}} \log p(\mathbf{w} \mid \boldsymbol{\mathcal{T}}) = \alpha\mathbf{I} + \beta\mathbf{R}, \tag{27}$$

where $\mathbf{R}$ is the Hessian matrix of the sum squared error with respect to $\mathbf{w}$, i.e.

$$\mathbf{R} = \nabla_{\mathbf{w}}\nabla_{\mathbf{w}} \sum_{o=1}^{O} \| f(\boldsymbol{\Phi}_o, \mathbf{w}) - \mathbf{T}_o \|^2. \tag{28}$$

We can further approximate the inverse of $\mathbf{A}$ as[4]

$$\mathbf{A}^{-1} \approx \frac{1}{\alpha}\mathbf{I} - \frac{\beta}{\alpha^2}\mathbf{R}. \tag{29}$$

[2] For a detailed derivation see [Cancino Chacon et al., 2014].

[3] See Section 2.3.3 in [Bishop, 2006] for a detailed derivation of this result.

[4] This approximation holds if $\beta << \alpha$. See [Petersen and Pedersen, 2012].

Using a Taylor expansion of the model output around the $\mathbf{w}_{map;(r)nbm}$

$$\mathrm{vec}(f(\mathbf{\Phi}, \mathbf{w})) \approx \mathrm{vec}(f(\mathbf{\Phi}, \mathbf{w}_{map;(r)nbm})) + \mathbf{J}_{map}(\mathbf{w} - \mathbf{w}_{map;(r)nbm}), \tag{30}$$

where $\mathbf{J}_{map}$ is the Jacobian matrix of the model output with respect to the parameters $\mathbf{w}$ evaluated in $\mathbf{w}_{map;(r)nbm}$. We can then write an approximation of the conditional probability $p(\mathbf{T} \mid \mathbf{w})$ that is Gaussian, and whose mean is a linear function of $\mathbf{w}$ as

$$p(\mathbf{T} \mid \mathbf{w}) \approx \mathcal{N}(\mathrm{vec}(\mathbf{T}) \mid \mathbf{m}_{lin}, \beta^{-1}\mathbf{I}), \tag{31}$$

with

$$\mathbf{m}_{lin} = \mathrm{vec}(f(\mathbf{\Phi}, \mathbf{w}_{map;(r)nbm})) + \mathbf{J}_{map}(\mathbf{w} - \mathbf{w}_{map;(r)nbm}). \tag{32}$$

These two approximations allow us to approximate the integral in Equation (21) in a similar way as in the linear case as

$$p(\mathbf{T} \mid \boldsymbol{\mathcal{T}}) = \mathcal{N}\left(\mathbf{T} \mid f(\mathbf{\Phi}, \mathbf{w}_{map;(r)nbm}), \sigma_N^2\right), \tag{33}$$

with

$$\sigma_N^2 = \frac{1}{\beta}\mathbf{I} + \mathbf{J}_{map}^{\mathrm{T}}\left(\frac{1}{\alpha}\mathbf{I} - \frac{\beta}{\alpha^2}\mathbf{R}\right)\mathbf{J}_{map}. \tag{34}$$

From the above results it follows that the linear-Gaussian approximation to compute the predictive distribution leads to a unimodal distribution.

(2) *Use Monte Carlo integration.* We can reformulate the computation of the predictive distribution from Equation (21) as calculating the expectation value of the conditional probability of $\mathbf{T}$ given the parameters $\mathbf{w}$ as

$$p(\mathbf{T} \mid \boldsymbol{\mathcal{T}}) = \mathbb{E}\left\{p(\mathbf{T} \mid \mathbf{w})\right\} \tag{35}$$

We can use *importance sampling*, a Monte Carlo technique for approximating expectation values [Bishop, 2006]. The posterior distribution $p(\mathbf{w} \mid \boldsymbol{\mathcal{T}})$ can be rewritten as

$$p(\mathbf{w} \mid \boldsymbol{\mathcal{T}}) = \frac{\overbrace{\tilde{p}(\mathbf{w} \mid \boldsymbol{\mathcal{T}})}^{p(\boldsymbol{\mathcal{T}}|\mathbf{w})p(\mathbf{w})}}{\underbrace{Z_p}_{\int p(\boldsymbol{\mathcal{T}}|\mathbf{w})p(\mathbf{w})d\mathbf{w}}}, \tag{36}$$

where Z_p is a normalization constant known as the *partition function.* Since generating samples from $p(\mathbf{w} \mid \boldsymbol{\mathcal{T}})$ is complicated, but evaluating $\tilde{p}(\mathbf{w} \mid \boldsymbol{\mathcal{T}})$ is straightforward, we introduce a *proposal distribution* $q(\mathbf{w} \mid \boldsymbol{\mathcal{T}})$ from which it is easy to draw samples. A candidate distribution for $q(\mathbf{w} \mid \boldsymbol{\mathcal{T}})$ is given by the Laplace approximation in Equation (26). Furthermore, we can rewrite this proposal distribution to match the form of Equation (36) as

$$q(\mathbf{w} \mid \boldsymbol{\mathcal{T}}) = \frac{\tilde{q}(\mathbf{w} \mid \boldsymbol{\mathcal{T}})}{Z_q}. \tag{37}$$

Using this, we can rewrite the expectation in Equation (35) as

$$\begin{aligned}\mathbb{E}\left\{p(\mathbf{T} \mid \mathbf{w})\right\} &= \int p(\mathbf{T} \mid \mathbf{w}) p(\mathbf{w} \mid \boldsymbol{\mathcal{T}}) d\mathbf{w} \\ &= \frac{Z_q}{Z_p} \int p(\mathbf{T} \mid \mathbf{w}) \frac{\tilde{p}(\mathbf{w} \mid \boldsymbol{\mathcal{T}})}{\tilde{q}(\mathbf{w} \mid \boldsymbol{\mathcal{T}})} q(\mathbf{w} \mid \boldsymbol{\mathcal{T}}) d\mathbf{w} \end{aligned} \tag{38}$$

By taking L samples from $\mathbf{w}_i^* \sim q(\mathbf{w} \mid \boldsymbol{\mathcal{T}})$, we can approximate the above expectation as

$$\mathbb{E}\left\{p(\mathbf{T} \mid \mathbf{w})\right\} \approx \frac{Z_q}{Z_p} \frac{1}{L} \sum_{l=1}^{L} r_l p(\mathbf{T} \mid \mathbf{w}_l^*), \tag{39}$$

where r_l is the l-th *importance weight*, given by

$$r_l = \frac{\tilde{p}(\mathbf{w}_l^* \mid \boldsymbol{\mathcal{T}})}{\tilde{q}(\mathbf{w}_l^* \mid \boldsymbol{\mathcal{T}})}. \tag{40}$$

Finally, we can approximate the ratio of partition functions $\frac{Z_p}{Z_q}$ as

$$\frac{Z_p}{Z_q} \approx \frac{1}{L} \sum_{l=1}^{L} r_l. \tag{41}$$

The computation of the predictive distribution using importance sampling is summarized in Algorithm 1.

```
Data:
$\boldsymbol{\mathcal{X}}$: set of training scores
$\boldsymbol{\mathcal{T}}$: set of training expressive targets
$\boldsymbol{\Phi}$: score to be rendered
$q(\mathbf{w} \mid \boldsymbol{\mathcal{T}})$: proposal posterior distribution
Generate a set of $L$ samples from the proposal distribution as
    $\mathbf{W}^* = \{\mathbf{w}_1^*, \dots, \mathbf{w}_L^* \mid \mathbf{w}_i^* \sim q(\mathbf{w} \mid \boldsymbol{\mathcal{T}})\ \forall i \in [1, L]\}$   (42)
Compute the set of importance weights as
    $\mathbf{r} = \left\{r_1, \dots, r_L \mid r_i = \frac{\tilde{p}(\mathbf{w}_i^* \mid \boldsymbol{\mathcal{T}})}{\tilde{q}(\mathbf{w}_i^* \mid \boldsymbol{\mathcal{T}})},\ \forall \mathbf{w}_i^* \in \mathbf{W}^*\right\}$   (43)
Compute the normalized importance weights as
    $\boldsymbol{w} = \left\{w_1, \dots, w_L \mid w_i = \frac{r_l}{\sum_{l=1}^{L} r_l},\ \forall r_i \in \mathbf{r}\right\}$   (44)
return Predictive distribution
    $p(\mathbf{T} \mid \boldsymbol{\mathcal{T}}) \approx \sum_{l=1}^{L} w_l p(\mathbf{T} \mid \mathbf{w}_l^*)$   (45)
```

Algorithm 1: Approximation of the predictive distribution using importance sampling.

Using the Gaussian assumptions from Equations (11) and (14), it is straightforward to see that the predictive distribution from Equation (45) is (possibly) multimodal, since it is a mixture of multivariate Gaussian distributions [Carreira-Perpiñán and Williams, 2003].

5. Conclusions

In this report, we used a Bayesian framework to produce a probabilistic interpretation for the different versions of the BM approach for generating expressive music performances. Additionally, derivations of the predictive distributions for the LBM and (R)NLBM under the assumption of Gaussian priors was provided. Future work could involve the exploration of more sophisticated Monte Carlo sampling procedures, such as Markov Chain Monte Carlo techniques, as well as variational methods. An alternative to the Bayesian framework proposed in this report would be the use of Mixture Density Networks [Bishop, 1995], where the outputs of a neural network represent the parameters of a probability distribution. These methods have been successfully applied to generation of sequences using RNNs [Graves, 2013]. Furthermore, it would be interesting to use more robust training methods for Bayesian Neural Networks like Probabilistic Backpropagation [Hernández-Lobato and Adams, 2015].

Acknowledgments

This work is supported by European Union Seventh Framework Programme, through the Lrn2Cre8 project (FET grant agreement no. 610859).

References

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Clarendon Press Oxford.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer Verlag, Microsoft Research Ltd.

Cancino Chacón, C. E. and Grachten, M. (2015). An evaluation of score descriptors combined with non-linear models of expressive dynamics in music. In Japkowicz, N. and Matwin, S., editors, *Proceedings of the 18th International Conference on Discovery Science (DS 2015)*, Lecture Notes in Artificial Intelligence, Banff, Canada. Springer.

Cancino Chacon, C. E., Grachten, M., and Widmer, G. (2014). Bayesian linear basis models with gaussian priors for musical expression. Technical report.

Carreira-Perpiñán, M. Á. and Williams, C. K. I. (2003). On the Number of Modes of a Gaussian Mixture. In *Scale-Space Methods in Computer Vision*, pages 625–640. Springer-Verlag, Berlin, Heidelberg.

Dauphin, Y. N., de Vries, H., Chung, J., and Bengio, Y. (2015). RMSProp and equilibrated adaptive learning rates for non-convex optimization. *arXiv*, 1502:4390.

Grachten, M. and Cancino Chacón, C. E. (2016). Temporal dependencies in the expressive timing of classical piano performances. Submitted.

Grachten, M., Cancino Chacón, C. E., and Widmer, G. (2014). Analysis and prediction of expressive dynamics using Bayesian linear models. In *Proceedings of the 1st international workshop on computer and robotic Systems for Automatic Music Performance*, pages 545–552.

Grachten, M. and Widmer, G. (2012). Linear Basis Models for Prediction and Analysis of Musical Expression. *Journal of New Music Research*, 41(4):311–322.

Graves, A. (2013). Generating Sequences With Recurrent Neural Networks. *arXiv*, 1308:850.

Hernández-Lobato, J. M. and Adams, R. P. (2015). Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks. *arXiv*, pages 1–10.

Petersen, K. B. and Pedersen, M. S. (2012). *The Matrix Cookbook* . Technical University of Denmark.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(9):533–536.