Supplementary Material for "Learning Similarity Metrics for Melody Retrieval"

Folgert Karsdorp, Peter van Kranenburg, Enrique Manjavacas
folgert.karsdorp@meertens.knaw.nl

1 Feature Details

We represent a melody as a sequence of notes, and each note as a set of feature-values. For each note, the following features are included:

Name	Range	Description
pitch	0...127	MIDI note number corresponding to a key of the piano keyboard. The value 60 corresponds to the middle C.
scaledegree	1...7	The scale degree of the pitch in the context of the key. Alterations are discarded.
duration	float	Duration of the note. Normalized by the length of a quarter note.
beat	float	The position of the note in the measure in units of the beat according to the meter model as implemented in python library music21 [1].
beatstrength	float	The metric weight of the position in the measure according to the meter model as implemented in music21.
metriccontour	['-', '=', '+']	Whether the beatstrength of the note is lower, equal or higher than that of the previous note. The first note gets '+'.
imaweight	float	The metric weight according to the Inner Metric Analysis [2].
imacontour	['-', '=', '+']	Whether the imaweight is lower, equal or higher than that of the previous note. The first note gets '+'.
phrasepos	float	The horizontal position of the note in its phrase; the onset time of the first note gets value 0.0, and the onset time of the final note gets value 1.0.

2 RNN Hyper-Parameters

The RNNs described in Section 3 of the paper consist of a large number of hyper-parameters, which were optimized through a randomized search on the development data set. Hyper-parameters were sampled according to the following criteria and distributions:

Name	**Description**
RNN Cell	Random choice between RNN Cells: $\{\text{GRU}, \text{LSTM}\}$
E	The embedding dimensionality E was sampled from distribution of even integers roughly following a truncated Normal distribution with $\mu = 8$, $\sigma = 3$, $4 < mu < 16$
H	The dimensionality of the hidden layer H was randomly chosen from: $\{64, 128, 256\}$;
n layers	The number of hidden layers randomly alternated between 1, 2, and 3;
bidirectional	Random choice of employing a bidirectional or unidirectional RNN: $\{0, 1\}$;
Embedding Dropout	The amount of embedding dropout was sampled from a truncated Normal distribution with $\mu = 0.5$, $\sigma = 0.1$, $0 < mu < 1$;
Layer Dropout	The amount of dropout between layers was samples from a truncated Normal distribution with $\mu = 0$, $\sigma = 0.1$, $0 \geq mu < 0.5$;
β	The positive loss scaling parameter β was sampled from a truncated Normal distribution with $\mu = 0.5$, $\sigma = 0.3$, $0.1 < mu < 1$;
α	The margin α was sampled from a truncated Normal distribution with $\mu = 0.5$, $\sigma = 0.3$, $0.1 < mu < 1$;
Loss variant	Random choice between hard and soft margin loss: $\{0, 1\}$;
η	The learning rate η was sampled from a truncated Normal distribution with $\mu = 0.001$, $\sigma = 0.001$, $0.0001 < mu < 0.01$;
Loss	Random choice between loss types: $\{\text{duplet}, \text{triplet}\}$;
Mini-batch classes	The number of unique classes per mini-batch was randomly chosen from $\{5, 10, 20\}$;
Mini-batch samples	The number of samples per class per mini-batch was randomly chosen from $\{5, 10, 20\}$;

Table 1 lists the hyper-parameters for the best performing models. Here, E refers to the dimensionality of the embedding space for categorical features, H is the dimensionality of the hidden layers, D_E refers to the amount of dropout applied on the input to the recurrent layer, D_L is the amount of dropout applied *between* recurrent layers, β refers to the scaling parameter of the positive loss, α is the margin, L refers to either 'hard' or 'soft' negative loss η refers to the learning rate, n is the number of unique classes per mini-batch, and, finally, m is the number of samples per class in each mini-batch.

	Cell	E	H	layers	bidir.	D_E	D_L	β	α	L	η	n	m
RNN_D	GRU	8	256	3	yes	.26	.056	.406	.606	hard	1.6e-03	20	5
RNN_T	GRU	8	256	2	yes	.37	.021	NA	.226	NA	4.5e-04	5	20

Table 1: Hyper-parameter settings for the best performing RNNs trained with duplet (RNN_D) and triplet (RNN_T) loss.

3 Linear Model Details

For brevity, details concerning the convergence of the Bayesian Linear model were left undiscussed in Section 4 of the paper. This section provides additional details about the convergence of the NUTS sampler. The full specification of the linear model is as follows:

$$
\begin{aligned}
\text{MAP}_i &\sim \text{Normal}(\mu_i, \sigma) \\
\mu_i &= \gamma + \beta_l l_i + \beta_m m_i + \beta_h h_i + \\
&\quad \beta_d d_i + \beta_b b_i + \beta_c c_i + \beta_{lm} l_i m_i \\
\gamma &\sim \text{Normal}(0, 1) \\
\beta_l &\sim \text{Normal}(0, 1) \\
\beta_m &\sim \text{Normal}(0, 1) \\
\beta_h &\sim \text{Normal}(0, 1) \\
\beta_d &\sim \text{Normal}(0, 1) \\
\beta_b &\sim \text{Normal}(0, 1) \\
\beta_c &\sim \text{Normal}(0, 1) \\
\beta_{lm} &\sim \text{Normal}(0, 1) \\
\sigma &\sim \text{Half-Cauchy}(0, 1)
\end{aligned}
\tag{1}
$$

Table 2 provides the posterior distribution estimates for the hyper-parameters of the Neural Networks with additional information about effective sample size and credible intervals.

Parameter	$\hat{R}$	n_eff	mean	sd	2.5%	50%	97.5%
γ	1.00	4000	0.66	0.00	0.65	0.66	0.67
β_l	1.00	4000	-0.08	0.00	-0.09	-0.08	-0.07
β_d	1.00	4000	-0.05	0.02	-0.09	-0.05	0.00
β_m	1.00	3894	0.01	0.01	-0.02	0.01	0.04
β_b	1.00	4000	0.03	0.01	0.01	0.03	0.04
β_c	1.00	4000	-0.05	0.00	-0.06	-0.05	-0.04
β_h	1.00	4000	0.02	0.00	0.01	0.02	0.03
$\beta_l m$	1.00	3296	-0.18	0.02	-0.22	-0.18	-0.13
σ	1.00	4000	0.04	0.00	0.04	0.04	0.04

Table 2: Posterior distribution estimates for the hyper-parameters of the Neural Networks. In addition to the mean estimates, the table provides the estimation errors, lower, median and upper 95% Credible Intervals, the $\hat{R}$ statistic, and the effective sample size.

The $\hat{R}$ scores are all well below 1.1 indicating good convergence and mixing of the chains. This is also reflected in the trace plots shown in Figure 1. The plots show healthy Markov chains that are stationary and well-mixing.

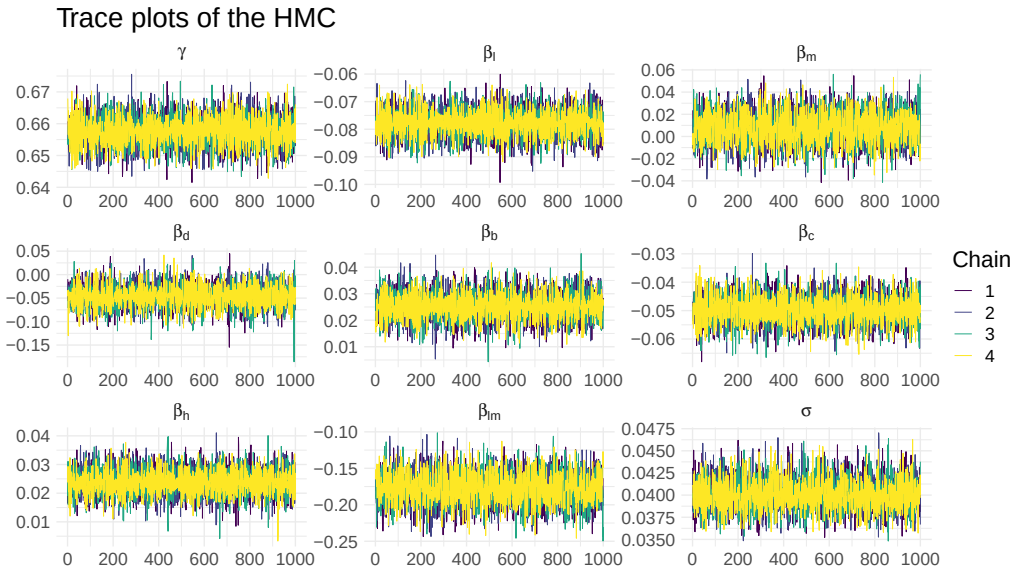

Figure 1: Trace plots of the Markov chain from the linear model.

As an additional validation of the posterior distribution estimates presented in Table 2, Figure 2 visualizes the complete posterior distribution for each of the predictors as well as the Intercept γ and σ.

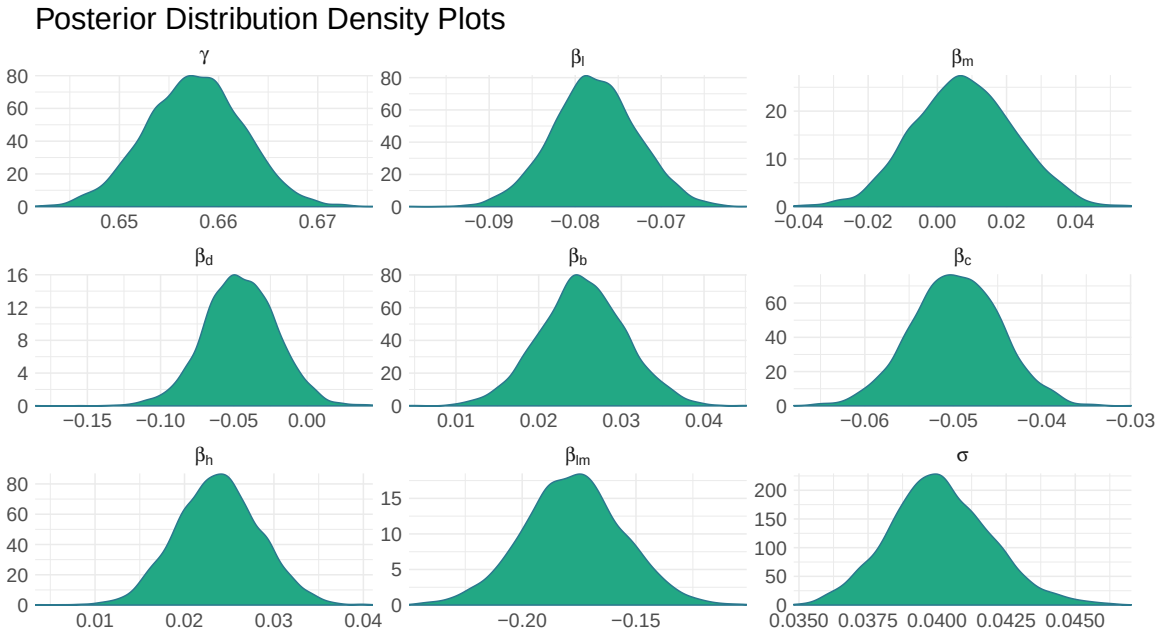

Figure 2: Posterior distribution density plots for all parameters in the linear model.

4 Silhouette Coefficient

The Silhouette Coefficient contrasts the mean similarity between a sample and all other samples from the same family with the similarity of that sample with members of other families:

$$\text{Silhouette}(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \text{ if } |C_i| > 1, \tag{2}$$

where $a(i)$ represents the mean distance between a tune family member and all other members from that family:

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} D(f(x_i), f(x_j)), \tag{3}$$

and $b(i)$ refers to the mean distance between a member of a tune family and all members of the nearest cluster from a different tune family:

$$b(i) = \min_{i \neq j} \frac{1}{|C_j|} \sum_{j \in C_j} D(f(x_i), f(x_j)) \tag{4}$$

By taking the average over all silhouette scores, we obtain a measure of cluster homogeneity ranging from -1 (incorrect clustering) to 1 (perfect clustering).

References

[1] Michael Scott Cuthbert and Christopher Ariza. Music21: A toolkit for computer-aided musicology and symbolic music data. In *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR 2010)*, pages 637–642, 2010.

[2] Anja Volk. Persistence and change: Local and global components of metre induction using inner metric analysis. *Journal of Mathematics and Computation in Music*, 2(2):99–115, 2008.