USING HYPER-GENRE TRAINING TO EXPLORE GENRE INFORMATION FOR AUTOMATIC CHORD ESTIMATION

Yizhao Ni, Matt Mcvicar, Raúl Santos-Rodríguez and Tijl De Bie
Intelligent Systems Laboratory
University of Bristol, U. K.
{enxyn, matt.mcvicar, enrsr, tijl.debie}@bristol.ac.uk

ABSTRACT

Recently a large amount of new chord annotations have been made available. This raises hopes for further development in automatic chord estimation. While more data seems to imply better performance, a major challenge however, is the wide variety of genres covered by these new data. As a result, the genre-independent training scheme as is common today is bound to fail. In this paper we investigate various options for exploring genre information for chord estimation, while also maximally exploiting the full dataset. More specifically, we propose a hyper-genre training scheme in which each genre cluster has its own parameters, tied together by hyper parameters as a Bayesian prior. The results are promising, showing significant improvements over other prevailing training schemes.

1. INTRODUCTION

Identifying musical chords from audio recordings is a challenging task and has recently attracted the interest of many researchers in the *music information retrieval* (MIR) field. The general approach of *automatic chord estimation* (ACE) involves two stages: the extraction of spectral features such as chromagram from audio; and the estimation of chords based on these features, via e.g. *Hidden Markov Models* (HMMs). In the past few years, while developing chroma extraction techniques has become a fruitful topic [2, 7–9], researchers have also explored a variety of musical factors such as key [5, 6, 10] and bassline [7, 9] that are related to chord progressions to build up richer ACE systems.

Nevertheless, one issue that has cramped the development in automatic chord estimation is the limited amount of the data available. Since most of the studies so far were carried out on a collection of The Beatles, Queen and Zweieck songs (i.e. the MIREX dataset), it is becoming increasingly probable that the existing ACE researches are overfitting this dataset. Recently a large amount of new chord annotations have been released by the *structural analysis of large amounts of music information* (SALAMI) project [14], raising hopes for further development of the ACE systems. A major challenge it also brings however, is the wide variety of genres covered by the data.

This paper is devoted to the study of the new dataset. Distinct from feature extraction and decoding research, we investigate various *training schemes* for exploring genre information to aid automatic chord estimation. We begin by giving an overview of the ACE task.

1.1 Automatic Chord Estimation

Let $\mathbf{x} = [x_1, \ldots, x_s, \ldots, x_S]$ be a mono audio signal with x_s indicating the value of the s-th sample. In ACE research, the signal is usually converted into a 12-dimensional representation of the harmonic content, with one such vector for each time frame. This vector is known as a *chroma* [2] vector, and it is intended to reflect the distribution of salience over the 12 pitch classes. The chroma vectors for the audio signal $\mathbf{x}$ are then gathered as the columns of a matrix $\mathbf{X} \in \mathbb{R}^{d \times T}$, with T denoting the number of frames and $d = 12$. In the target domain the chord annotations are denoted by $\mathbf{c} \in \mathcal{A}^{1 \times T}$, with $\mathcal{A}$ representing the chord alphabet. To model the relationship between observed variables $\mathbf{X}_t$ and hidden variables c_t, a standard HMM [13] with the parameter set $\Theta = \left\{\mathbf{P}_i \in \mathbb{R}^{|\mathcal{A}|}, \mathbf{P}_t \in \mathbb{R}^{|\mathcal{A}| \times |\mathcal{A}|}, \mathbf{P}_e\right\}$ is commonly used. $\mathbf{P}_i$, $\mathbf{P}_t$ denote the initialization and the transition probabilities respectively, and the emission probability for chord c_t is frequently modelled as a single Gaussian

$$p_e(\mathbf{X}_t|c_t) = \mathbf{X}_t \sim \mathcal{N}(\boldsymbol{\mu}^{c_t}, \boldsymbol{\Sigma}^{c_t}) \tag{1}$$

with the distribution parameters $\{\boldsymbol{\mu}^c, \boldsymbol{\Sigma}^c\}_{c \in \mathcal{A}}$. Under this framework, the joint probability of the feature vectors $\mathbf{X}$ and the corresponding chord sequence $\mathbf{c}$ is of the form

$$P(\mathbf{X}, \mathbf{c}|\Theta) = p_i(c_1) \prod_{t=2}^{T} p_t(c_t|c_{t-1}) \prod_{t=1}^{T} p_e(\mathbf{X}_t|c_t). \tag{2}$$

Given the optimal parameters Θ^*, the ACE task is equivalent to finding $\mathbf{c}^*$ that maximizes the joint probability $\mathbf{c}^* = \arg\max_{\bar{\mathbf{c}}} P(\mathbf{X}, \bar{\mathbf{c}}|\Theta^*)$, which can be done efficiently by the Viterbi algorithm [13].

By restricting our interest to a standard HMM, the *training scheme* as we define it is a strategy to derive the optimal parameters Θ^* from the training data. Given N audio clips for which the chromagrams $\mathcal{X} = \{\mathbf{X}^n \in \mathbb{R}^{d \times T_n}\}_{n=1}^{N}$ and

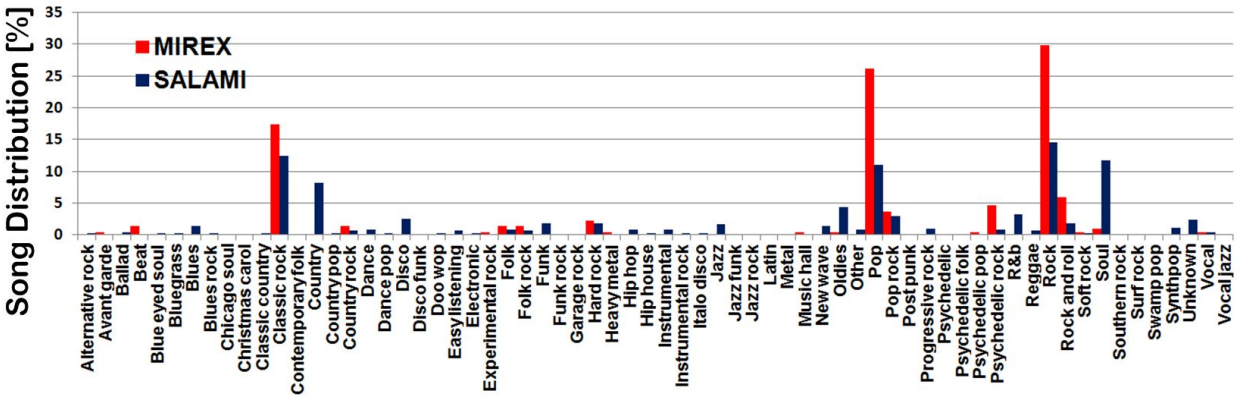

Figure 1. Genre distribution of the MIREX and the SALAMI datasets. The MIREX dataset is dominated by Rock and Pop genres, whereas the SALAMI dataset has a much wider variety of genres such as Country and Blues/Soul.

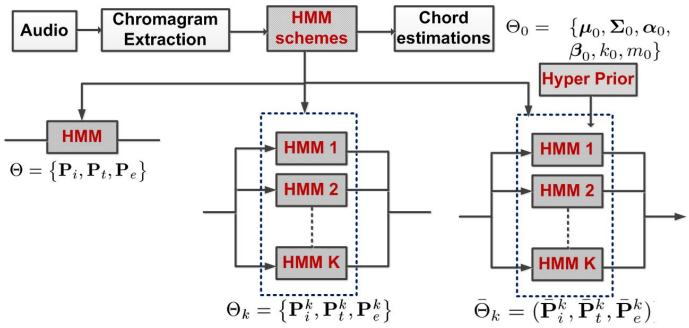

Figure 2. Training schemes for automatic chord estimation: universal training (left), genre-specific training (middle) and the proposed hyper-genre training (right).

the chord annotations $\mathcal{C} = \{\mathbf{c}^n \in \mathcal{A}^{1\times T_n}\}_{n=1}^N$ are both available, the prevailing scheme is the *universal training* (denoted by UN, cf. left block in Fig. 2) that derives one Θ^* from all available data $\{\mathcal{X}, \mathcal{C}\}$.

1.2 Why a new training scheme?

The effectiveness of UN-training on the MIREX dataset has now been established. Since this collection is highly genre-biased and only small variations exist (cf. Fig. 1), UN-training can make full use of the data without confounding chord characteristics. However, the SALAMI data has a much wider variety of genres [1] (cf. Fig. 1), this casts doubts on the effectiveness of UN-training for two reasons: first of all, when reducing the chords to an alphabet such as triads (cf. Fig. 3), the variety of chords is noticeably different between genres. For instance, Rock and Country genres mainly use simple chords (e.g. major), whereas Jazz tends to uses more complex ones (e.g. major 7th). This subsequently incurs a large variation on the chromas of the same chord among different genres (cf. Fig.4). In addition, chord progressions of different genres can vary dramatically [11]. Neither of these can be solved by UN-training, since the scheme ignores the connection between musical genre and chord variety and progression.

[1] The genre information was obtained with thanks from `http://www.last.fm/` and `http://www.wikipedia.org/`.

A potential solution to these issues is to apply a more complex model such as a Mixture of Gaussians (MOG) to Eqn. (1), but this risks the probability functions of different chords being confused [e.g. E:dim= (E, G, Bb) may also yield high probability in an MOG model for C:maj due to the modelling of C:7= (C, E, G, Bb)]. One can also respect this musical factor with a more rigorous approach: training a different Θ for each genre, an example of which is the *genre-specific training* (denoted by GS, cf. middle block in Fig.2) presented in [5]. However, this method often suffers from the problems caused by data sparseness, because it can not maximally exploit the full dataset (see the discussion in Sec. 3.1).

As an alternative, we develop a new training scheme which we call *hyper-genre training* (denoted by HG, cf. right block in Fig. 2). Instead of using independent parameters for each genre as it is in GS-training, HG-training constructs a hierarchical probabilistic model and connect the genres using hyper parameters. This framework is similar to a Hierarchical Dirichelet Process (HDP) [15], which has been applied to music similarity measurement [12] and timbral similarity estimation [4] in the MIR domain. The main difference here is that the proposed approach uses a well-defined cluster structure on the basis of musical knowledge, hence avoiding the massive sampling and uncertain clustering process in HDP.

The rest of the paper is organized as follows. In Sec. 2 we apply the proposed HG-training to the standard HMM and detail the corresponding parameter estimation. We then evaluate the approach and compare it with the other two training schemes in Sec. 3. Finally the conclusions and future work are drawn in Sec. 4.

2. HYPER-GENRE HMM

To take into account the fact that songs belong to different genres, the data is divided into K clusters according to the genre information. Suppose the k-th cluster contains n_k songs and $\sum_{k=1}^K n_k = N$, we then denote the collection of chromagrams and chord annotations for cluster k as $\mathcal{X}_k = \{\mathbf{X}^n \in \mathbb{R}^{d\times T_n}\}_{n=1}^{n_k}$, and $\mathcal{C}_k = \{\mathbf{c}^n \in \mathcal{A}^{1\times T_n}\}_{n=1}^{n_k}$.

In general, a UN-HMM trains one Θ on all available

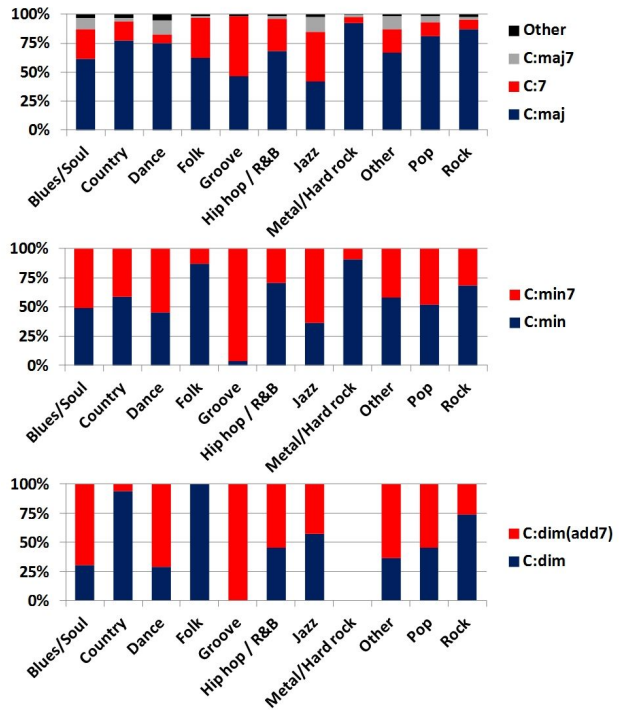

Figure 3. Comparison of chord varieties of chord C between different genres. Each figure includes the chords that can be reduced to the same triad. The percentages of these chords in a cluster then suggest the genre-specific chord variety. Note that in this figure the genres appeared in the SALAMI dataset have been grouped manually and formed 11 genre clusters.

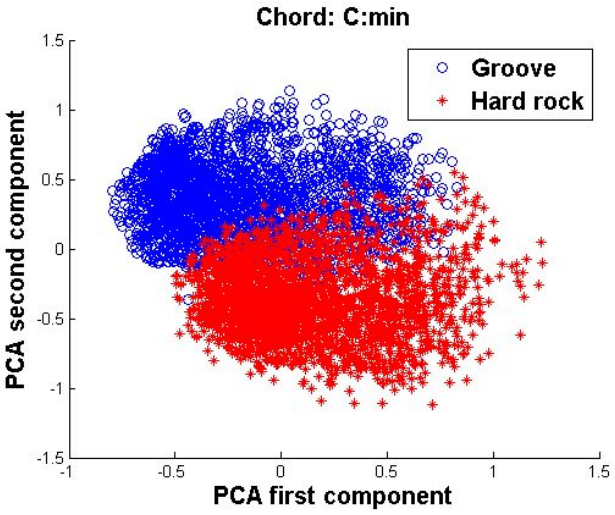

Figure 4. Chroma features for all occurrences of C:min-like chords (cf. middle plot in Fig. 3) for the Groove and the Hard rock genres. To aid visualization, the 12 dimensional feature space has been reduced to 2 using Principal Component Analysis. We found that in the Groove cluster, there were many more complex variants (e.g. C:min7), whilst in Hard rock simple chords such as C:min were more common. Owing to this, there is a large variation between their chroma features.

data $\{\mathcal{X}, \mathcal{C}\}$; while a GS-HMM has a set of parameters $\mathbf{\Theta} = \{\Theta_k\}_{k=1}^K$, each of which is trained on the cluster examples $\{\mathcal{X}_k, \mathcal{C}_k\}$. The HG-HMM also has a set of genre-specific parameters $\bar{\mathbf{\Theta}} = \left\{\bar{\Theta}_k = (\bar{\mathbf{P}}_i^k, \bar{\mathbf{P}}_t^k, \bar{\mathbf{P}}_e^k)\right\}_{k=1}^K$, but it ties them together by hyper parameters as a Bayesian prior.

One implementation of the HG-HMM is depicted in Fig. 5, in which the genre clusters are connected via a hyper parameter set $\Theta_0 = \{\boldsymbol{\mu}_0 \in \mathbb{R}^{d\times|\mathcal{A}|}, \mathbf{\Sigma}_0 \in \mathbb{R}^{d\times d\times|\mathcal{A}|}, \boldsymbol{\alpha}_0 \in \mathbb{R}^{|\mathcal{A}|}, \boldsymbol{\beta}_0 \in \mathbb{R}^{|\mathcal{A}|\times|\mathcal{A}|}, \mathbf{k}_0 \in \mathbb{N}^K, m_0\}$ to propagate information. Under this framework, the parameter estimates of $\bar{\mathbf{\Theta}}$ are computed as follows.

2.1 Parameter Estimation

For the emission probability $\bar{\mathbf{P}}_e^k$ of cluster k, the hyper link (dash line in Fig. 5) is equivalent to applying a conjugate prior to the distribution parameters $\{\boldsymbol{\mu}_k^c, \mathbf{\Sigma}_k^c\}_{c\in\mathcal{A}}$:

$$\begin{aligned} &\mathbf{\Sigma}_k^c \sim \mathcal{W}(\mathbf{\Sigma}_0^c, m_0) \\ &\boldsymbol{\mu}_k^c|\mathbf{\Sigma}_k^c \sim \mathcal{N}(\boldsymbol{\mu}_0^c, \tfrac{1}{k_0^c}\mathbf{\Sigma}_k^c) \end{aligned} \quad \forall c \in \mathcal{A}, \tag{3}$$

where $\mathcal{W}$ denotes the Wishart distribution [1]. The Bayesian update of the emission probability then becomes

$$\begin{aligned} \bar{p}_e^k(\mathbf{X}_t|c) = \quad &\mathbf{X}_t \sim \mathcal{T}(m_0 + m_k^c - d + 1, \\ &\bar{\boldsymbol{\mu}}_k^c, \frac{k^c + 1}{k^c(m_0 + m_k^c - d + 1)}\bar{\mathbf{\Sigma}}_k^c). \end{aligned} \tag{4}$$

In (4) $\mathcal{T}$ denotes the multivariate Student-t distribution with the following parameters

$$\begin{aligned} &m_k^c = \#(c_t = c), \forall c_t \in \mathcal{C}_k, \\ &k^c = k_0^c + m_k^c, \\ &\bar{\boldsymbol{\mu}}_k^c = \tfrac{k_0^c\boldsymbol{\mu}_0^c + m_k^c\boldsymbol{\mu}_k^{*c}}{k_0^c + m_k^c}, \\ &\bar{\mathbf{\Sigma}}_k^c = \mathbf{\Sigma}_0^c + m_k^c\mathbf{\Sigma}_k^{*c} + \tfrac{k_0^c m_k^c}{k^c}\|\boldsymbol{\mu}_k^{*c} - \boldsymbol{\mu}_0^c\|^2, \end{aligned} \tag{5}$$

where $\#$ indicates 'the number of' and $\{\boldsymbol{\mu}_k^{*c}, \mathbf{\Sigma}_k^{*c}\}_{c\in\mathcal{A}}$ are the maximum likelihood (ML) estimations of the parameters $\{\boldsymbol{\mu}_k^c, \mathbf{\Sigma}_k^c\}_{c\in\mathcal{A}}$ using the cluster examples $\{\mathcal{X}_k, \mathcal{C}_k\}$.

Similarly, the hyper priors applied to the initialization and the transition parameters of cluster k are given by

$$\begin{aligned} &\mathbf{P}_i^k|\boldsymbol{\alpha}_0 = \{p_i^k(c)|\ c \in \mathcal{A}\} \sim \mathrm{Dir}(|\mathcal{A}|, \boldsymbol{\alpha}_0), \\ &\mathbf{P}_t^k|\boldsymbol{\beta}_0 = \{p_t^k(c|\bar{c})|\ c \in \mathcal{A}\} \sim \mathrm{Dir}(|\mathcal{A}|, \boldsymbol{\beta}_0^{\bar{c}}), \forall \bar{c}, \end{aligned} \tag{6}$$

where Dir is the Dirichlet distribution. The Bayesian update of these probabilities are then computed by: $\forall c_1, c_{t-1}, c_t \in \mathcal{C}_k$

$$\begin{aligned} &\bar{p}_i^k(c) = \frac{\#(c_1 = c) + \alpha_0^c}{\sum_{c'\in\mathcal{A}} \#(c_1 = c') + \sum_{c'\in\mathcal{A}} \alpha_0^{c'}}, \\ &\bar{p}_t^k(c|\bar{c}) = \frac{\#(c_t = c\ \&\ c_{t-1} = \bar{c}) + \beta_0^{\bar{c},c}}{\sum_{c'\in\mathcal{A}} \#(c_t = c'\ \&\ c_{t-1} = \bar{c}) + \sum_{c'\in\mathcal{A}} \beta_0^{\bar{c},c'}}. \end{aligned} \tag{7}$$

Note that if a non-informative prior is used (i.e. $\alpha_0^c = 1$ and $\beta_0^{\bar{c},c} = 1$), the Bayesian update (7) is the ML estimations of the initialization and the transition parameters trained on the cluster examples. This is equivalent to using $\{\mathbf{P}_i^k, \mathbf{P}_t^k\}_{k=1}^K$ as used in the GS-HMM.

Eqns. (5) and (7) provide the insight of the hyper parameters: they reflect our prior belief about the genre cluster parameters. Hence when few data are available to estimate the parameters of cluster k, the HG-HMM can benefit

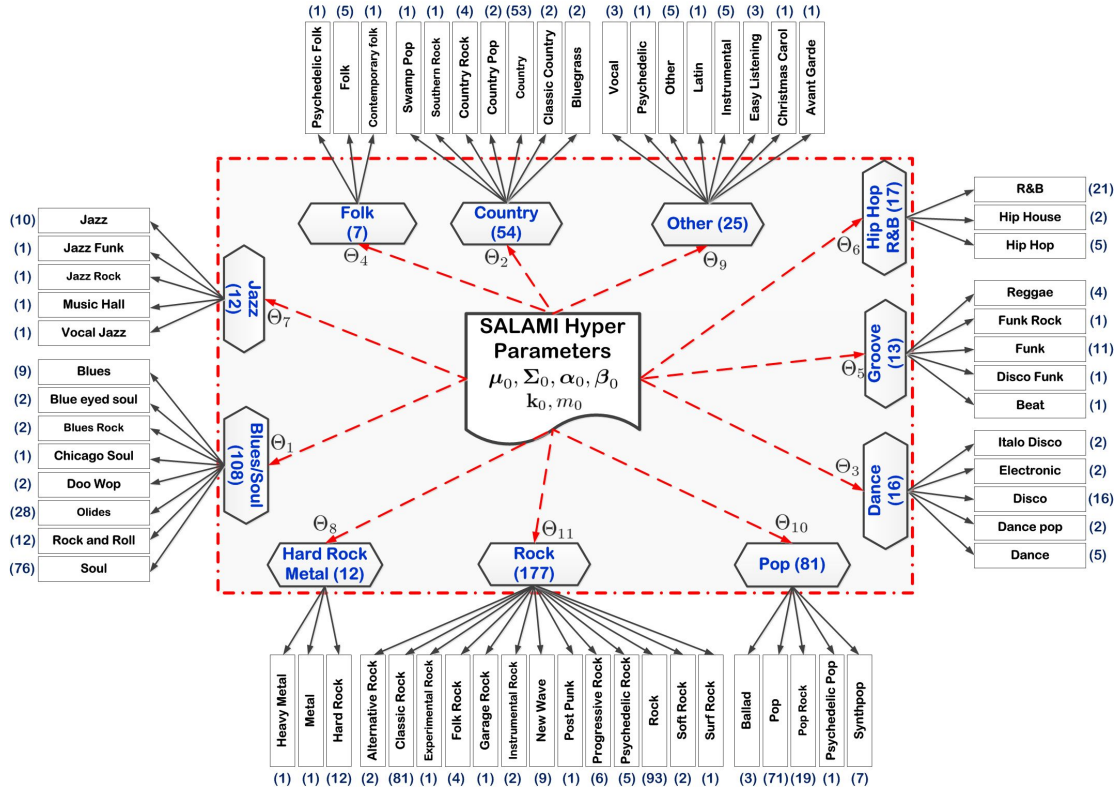

Figure 5. The implementation (dash-dot box) of the hyper-genre HMM for the SALAMI dataset. Ideally each genre should be regarded as a cluster, but in practice this is difficult to achieve with limited data. In order to assure a reasonable cluster size, we grouped the genres and created 11 genre-related clusters. These clusters are then connected via the hyper parameters so as to share information. The number of songs in a genre (or cluster) is shown in the bracket.

from the hyper parameter set. The clusters (e.g. Rock) can also share information with related ones (e.g. Blues), while retaining their intrinsic chord varieties and progressions. Ideally the set Θ_0 should has several subsets, one for each group of related genres. However, restricted by the data available we had to follow the suggestions in [4, 12] and used a single Θ_0 that reflects the distribution of the whole dataset instead: that is, $(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ are set to the mean and the covariance matrices of all training data respectively; $\boldsymbol{\alpha}_0, \boldsymbol{\beta}_0$ counts for all chord initializations and transitions, $m_0 = d$ and $k_0^c = \#(c_t = c), \forall c_t \in \mathcal{C}$.

2.2 Decoding

Given the updated parameters $\{\bar{\Theta}_k\}_{k=1}^K$, the decoding process is given by

$$\begin{aligned} \mathbf{c}^* &= \arg\max_{\bar{\mathbf{c}}} P(\mathbf{X}, \bar{\mathbf{c}}|\bar{\Theta}_k) \\ &= \arg\max_{\bar{\mathbf{c}}} \bar{p}_i^k(c_1) \prod_{t=2}^{T} \bar{p}_t^k(c_t|c_{t-1}) \prod_{t=1}^{T} \bar{p}_e^k(\mathbf{X}_t|c_t) \end{aligned} \tag{8}$$

The decoder (8) requires the cluster label k of the test example. This requirement can be easily waived by using the maximum likelihood inference as suggested in [5]:

$$\{\mathbf{c}^*, k^*\} = \arg\max_{\bar{\mathbf{c}}, k} P(\mathbf{X}, \bar{\mathbf{c}}|\bar{\Theta}_k). \tag{9}$$

3. EXPERIMENTS

Here we describe the main experiments conducted. The dataset investigated is the SALAMI data, which contains 522 songs along with the ground truth chord annotations [2]. In the experiments, we restrict the ACE system to a standard HMM with the loudness based chromagram [9]. In order to capture intrinsic chord varieties and progressions between genres but retaining a controllable complexity, we restricted ourselves to an alphabet of triads [3], with 73 unique chords in total. To evaluate the proposed approach, we randomly split $2/3$ of songs from each genre cluster to form the training set, while the remaining $1/3$ were used for testing. The frame-based chord estimation accuracy is used as the evaluation metric and in total 102 train-test runs were done to assess variance [4].

3.1 Comparison of different training schemes

There are three training schemes we can apply to the emission parameters: universal training (UN), genre-specific training (GS) and hyper-genre training (HG). Similarly, they can be applied to the initialization and the transition parameters, resulting in $3 \times 3 = 9$ combinations. Suppose the

[2] The readers are referred to the Appendix for our process of extracting the SALAMI chord annotations. These annotations are available online at `https://patterns.enm.bris.ac.uk/files/SALAMI_522_chord_annotations.zip`.

[3] Chord types: maj, min, dim, sus2, sus4, aug and N.

[4] That is, each song in the dataset would be tested 34 times.

cluster labels are known, in this subsection we compared these combinations using the decoder (8).

Table. 1 presents the overall performances for each combination, from which we observed a consistent improvement of the hyper-genre training over the other schemes. In particular, the HG-GS combination achieves the best performance, amounting to 9.3% and 14.1% reductions on the error rate compared with the universal (UN-UN) and the genre-specific (GS-GS) trainings respectively. Although applying a transition prior learnt from all the data still yields better results than the other schemes, it is worse than merely using a non-informative one. We postulate this is because a transition prior learnt from all examples is a mixed chord progression of all genres, applying it would inevitably confound some typical progressions such as "I (tonic) - V (dominant) - IV (subdominant)" commonly seen in the Blues genre. This suggests that further improvement might be gained by applying musical knowledge based transition priors on different genres, which can be obtained from e.g. the synthetic MIDI data used in [5].

E\T	UN	GS	HG
UN	$\boxed{63.61}\pm 1.31$	64.39 ± 1.29	64.3 ± 1.30
GS	60.61 ± 1.60	$\boxed{61.56}\pm 1.68$	61.05 ± 1.67
HG	65.93 ± 1.17	$\mathbf{66.98} \pm 1.21$	66.47 ± 1.19

Table 1. Performances $[\%]$ of different scheme combinations on the SALAMI dataset (best result in bold). The vertical axis shows the schemes applied to the emission parameters and the horizontal axis shows that to the initialization/transition parameters. The improvement of the HG-HMM with non-informative priors (HG-GS) is significant at a level $< 10^{-34}$ over the performances of the other scheme combinations under a paired t-test.

Figure 6 further depicts the performances of different training schemes on each genre cluster. For GS-training, the performance gradually increases when more examples are available for a cluster. The only exception is Groove, possibly due to the fact that the complex chords in this cluster are difficult to estimate. Limited by the amount of the data available for each cluster, GS-training is generally inferior to UN-training. This problem was not experienced or explored in [5] (Sec. 4.3), since their experiments were based on synthetical MIDI data such that the GS-HMM had sufficient examples to train the parameters. Alternatively, HG-training is slightly worse than UN-training when the data is very limited, by means of sharing information from the hyper parameters. For other clusters such as Rock, HG-training allows it to obtain information from related clusters such as Blues/Soul, while retaining the genre-specific chord variety and progression. This benefit makes it outperform UN-training and improve the performance by an absolute 4.4%.

3.2 Bypassing the genres

In practice the genre information of the test data is unknown, hence there is no choice but to use a genre-independent

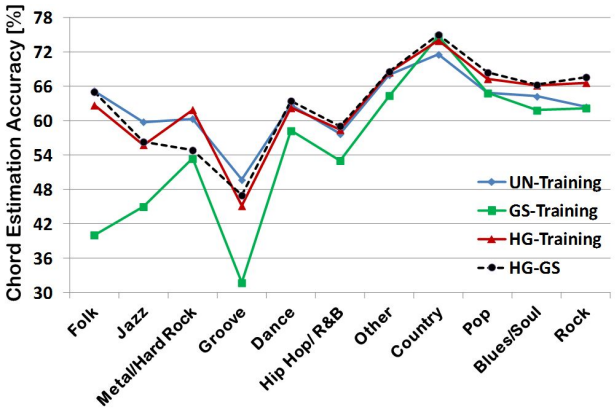

Figure 6. Performances of the universal (UN-UN), the genre-specific (GS-GS), the hyper-genre (HG-HG) and the combined (HG-GS) training schemes on each cluster. The clusters from left to right are sorted by the number of examples in the clusters.

model (e.g. UN-training); or increase the model complexity and infer the genre as well. In this subsection we investigate how much we can gain by inferring the genre with maximum likelihood technique (9). The experiment setup was the same as that in Sec. 3.1, and we compared the following models with the ones presented in Tab. 1:

1) The GS-HMM (using the schemes GS-GS) with the decoder (9). This is the model suggested in [5] (Sec. 4.3).

2) The HG-HMM (using HG-GS) with the decoder (9).

In addition to chord estimation accuracies, the genre prediction accuracies of the models are also evaluated.

Tab. 2 shows the results and we observed a mild decrease in performance when compared with the same models using genre information in Tab. 1. This reflects the benefit obtained from the genre information used. However, the performances of using ML inference are very close to that of using genre information directly, suggesting that this technique is reliable to use when the genre for a test example is unknown.

It is worth pointing out that the genre prediction accuracies are very low for both models, probably for two reasons. Firstly, since the numbers of examples in different clusters are highly imbalanced, some clusters might not have enough data to train the parameters (e.g. Folk and Jazz). In this case, a test example from that cluster could be better decoded by other cluster models. Although applying the hyper parameters can ease this problem and improve the chord estimation accuracy, it makes the cluster models closer to the hyper prior and inevitably confounds their intrinsic chord varieties. This consequently worsens the genre prediction accuracies.

Another potential explanation is that since the genre clusters are highly overlapped, it is very difficult to rigorously classify an example into just one cluster. This seems to suggest using a more flexible forest structure (e.g. the genre Country Rock should be connected to both Country and Rock clusters) instead of the directed tree depicted in Fig. 5, which will be investigated in our future work.

	GS-GS	HG-GS
C-Acc	61.27 ± 1.50	$\mathbf{66.83} \pm 1.27$
G-Acc	$\mathbf{16.05} \pm 2.98$	10.52 ± 1.81

Table 2. Performances [%] of different models on the SALAMI dataset (best result in bold). C-Acc denotes the frame-based chord estimation accuracy and G-Acc is the genre prediction accuracy.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a new training scheme – *hyper-genre training* for automatic chord estimation, capable of testing on multiple varied genres. The principle is to construct a hierarchical probabilistic model and connect the genre clusters using hyper parameters. Compared with the prevailing universal training scheme, HG-training is able to retain chord variety and progression characteristics of musical styles. Compared with genre-specific training, HG-training can benefit from the hyper prior and it resolves the problem of data sparseness often encountered in real world data. Both benefits have been verified in our experiments on a large and varied chord annotation dataset, where HG-training achieved significant improvements over the other two schemes.

For future work, we aim to improve the hierarchical structure of the proposed approach. This can be done by employing a more flexible forest structure instead of the directed tree graph. An alternative direction of research is to learn such hierarchical structure from the data automatically, which might lead to a more robust and powerful ACE system. Finally, we are also interested in how incorporating musical knowledge based transition priors may improve chord estimation accuracy.

5. APPENDIX: EXTRACTING CHORD ANNOTATIONS FOR THE SALAMI DATASET

In this appendix we summarize how we extracted the ground truths from the SALAMI chord annotation files.

There are two processes: obtaining the chord labels and inferring the durations. For the former, we followed the description in [14] and parsed the chord labels with the C. Harte's chord parser [3]. There were several exceptions, which we revised manually. A more difficult task is to extract chord durations. The SALAMI chord annotation files do not offer time stamps for each chord (as used in the MIREX annotation files). Instead, time instances are given over multiple bars, where each bar contains one or more chord. Our assumption is that the bars between two time stamps would have equal durations (if they are not in the same meter, then the durations are adjusted according to the meter). Under this assumption we extracted chord durations from the annotation files.

The original SALAMI dataset contains 649 songs, from which we found 21 songs having ambiguous tuning. Additionally there are some duplications and cover songs. For our experiments, we removed the 21 songs and the duplicate and cover songs so that each unique song only appeared once in the dataset. After this process we obtained a set of 522 songs.

6. REFERENCES

[1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[2] T. Fujishima. Real time chord recognition of musical sound: a system using common lisp music. In *Proc. of ICMC*, pages 464–467, 1999.

[3] C. Harte, M. Sandler, and S. Abdallah. Symbolic representation of musical chords: a proposed syntax for text annotations. In *Proc. of ISMIR*, pages 66–71, 2005.

[4] M. Hoffman. *Probabilistic graphical models for the analysis and synthesis of music audio*. PhD thesis, Princeton University, 2010.

[5] K. Lee. *A system for acoustic chord transcription and key extraction from audio using hidden Markov models trained on synthesized audio*. PhD thesis, Stanford University, 2008.

[6] K. Lee and M. Slaney. Acoustic chord transcription and key extraction from audio using key-dependent hmms trained on synthesized audio. *The IEEE Transactions on Audio, Speech and Language Processing*, 2008.

[7] M. Mauch. *Automatic chord transcription from audio using computational models of musical context*. PhD thesis, Queen Mary University of London, 2010.

[8] M. Müller and S. Ewert. Towards timbre-invariant audio features for harmony-based music. *IEEE Trans. Audio, Speech, Lang. Process.*, 18(3):649–662, 2010.

[9] Y. Ni, M. Mcvicar, R. Santos-Rodriguez, and T. De Bie. An end-to-end machine learning system for harmonic analysis of music. *IEEE Trans. Audio, Speech, Lang. Process.*, 20(5), 2012.

[10] K. Noland and M. Sandler. Key estimation using a hidden Markov model. In *Proc. of ISMIR*, 2006.

[11] W. Piston. *Harmony*. Norton, New York, 1978.

[12] Y. Qi, J. Paisley, and L. Carin. Music analysis using hidden markov mixture models. *IEEE Transactions on Signal Processing*, 55(11):5209–5224, 2007.

[13] L. R. Rabiner. A tutorial on hidden Markov models and selected application in speech recognition. In *Proc. of the IEEE*, 1989.

[14] J. Smith, J. Burgoyne, I. Fujinaga, D. Roure, and J. Downie. Design and creation of a large-scale database of structural annotations. In *Proc. of ISMIR*, 2011.

[15] Y. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2007.