

A PROBABILISTIC TOPIC MODEL FOR UNSUPERVISED LEARNING OF MUSICAL KEY-PROFILES

Diane J. Hu and Lawrence K. Saul

Department of Computer Science and Engineering
University of California, San Diego
{dhu, saul}@cs.ucsd.edu

ABSTRACT

We describe a probabilistic model for learning musical key-profiles from symbolic files of polyphonic, classical music. Our model is based on Latent Dirichlet Allocation (LDA), a statistical approach for discovering hidden topics in large corpora of text. In our adaptation of LDA, symbolic music files play the role of text documents, groups of musical notes play the role of words, and musical key-profiles play the role of topics. The topics are discovered as significant, recurring distributions over twelve neutral pitch-classes. Though discovered automatically, these distributions closely resemble the traditional key-profiles used to indicate the stability and importance of neutral pitch-classes in the major and minor keys of western music. Unlike earlier approaches based on human judgement, our model learns key-profiles in an unsupervised manner, inferring them automatically from a large musical corpus that contains no key annotations. We show how these learned key-profiles can be used to determine the key of a musical piece and track its harmonic modulations. We also show how the model's inferences can be used to compare musical pieces based on their harmonic structure.

1. INTRODUCTION

Musical composition can be studied as both an artistic and theoretical endeavor. Though music can express a vast range of human emotions, ideas, and stories, composers generally work within a theoretical framework that is highly structured and organized. In western tonal music, two important concepts in this framework are the *key* and the *tonic*. The key of a musical piece identifies the principal set of pitches that the composer uses to build its melodies and harmonies. The key also defines the tonic, or the most stable pitch, and its relationship to all of the other pitches in the key's pitch set. Though each musical piece is characterized by one overall key, the key can be shifted within a piece by a compositional technique known as *modulation*. Notwithstanding the infinite number of variations possible

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.

© 2009 International Society for Music Information Retrieval.

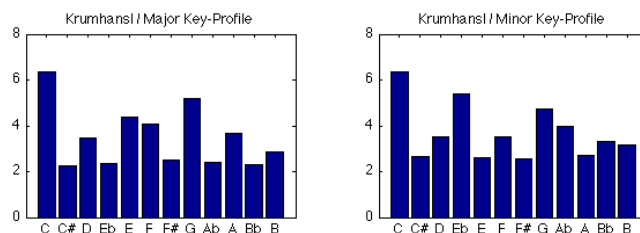


Figure 1. C major (left) and C minor (right) key-profiles proposed by Krumhansl-Kessler (KK), used in the Krumhansl-Schmuckler (KS) key-finding algorithm.

in music, most pieces can be analyzed in these terms.

Musical pieces are most commonly studied by analyzing their melodies and harmonies. In any such analysis, the first step is to determine the key. While the key is in principle determined by elements of music theory, individual pieces and passages can exhibit complex variations on these elements. In practice, considerable expertise is required to resolve ambiguities.

Many researchers have proposed rule-based systems for automatic key-finding in symbolic music [2, 10, 12]. In particular, Krumhansl and Schmuckler (KS) [8] introduced a model based on “key-profiles”. A key-profile is a twelve-dimensional vector in which each element indicates the stability of a neutral pitch-class relative to the given key. There are 24 key-profiles in total, one for each major and minor key. Using these key profiles, KS proposed a simple method to determine the key of a musical piece or shorter passages within a piece: first, accumulate a twelve-dimensional vector whose elements store the total duration of each pitch-class in a song; second, compute the key-profile that has the highest correlation with this vector. The KS model uses key-profiles derived from probe tone studies conducted by Krumhansl and Kessler (KK) [9]. Figure 1 shows the KK key profiles for C major and C minor; profiles for other keys are obtained by transposition. In recent work [14, 15], these key-profiles have been modified to achieve better performance in automatic key-finding.

In this paper, we show how to learn musical key-profiles automatically from the statistics of large music collections. Unlike previous studies, we take a purely data-driven approach that does not depend on extensive prior knowledge of music or supervision by domain experts. Based on a model of *unsupervised* learning, our approach bypasses the

need for manually key-annotated musical pieces, a process that is both expensive and prone to error. As an additional benefit, it can also discover correlations in the data of which the designers of rule-based approaches are unaware. Since we do not rely on prior knowledge, our model can also be applied in a straightforward way to other, non-western genres of music with different tonal systems.

Our approach is based on Latent Dirichlet Allocation (LDA) [1], a popular probabilistic model for discovering latent semantic topics in large collections of text documents. In LDA, each document is described as a mixture of topics, and each topic is characterized by its own particular distribution over words. LDA for text is based on the premise that documents about similar topics contain similar words. Beyond document modeling, LDA has also been adapted to settings such as image segmentation [5], part-of-speech tagging [6], and collaborative filtering [11].

Our variant of LDA for unsupervised learning of key-profiles is based on the premise that musical pieces in the same key use similar sets of pitches. Roughly speaking, our model treats each song as a “document” and the notes in each beat or half-measure as a “word”. The goal of learning is to infer harmonic “topics” from the sets of pitches that commonly co-occur in musical pieces. These harmonic topics, which we interpret as key-profiles, are expressed as distributions over the twelve neutral pitch-classes.

We show how to use these key-profiles for automatic key-finding and similarity ranking of musical pieces. We note, however, that our use of key-profiles differs from that of the KS model. For key-finding, the KS model consists of two steps: 1) derive key-profiles and 2) predict keys using key-profiles. In our model, these steps are naturally integrated by the Expectation-Maximization (EM) algorithm [3]. We do not need further heuristics to make key-finding predictions from our key-profiles as the EM algorithm yields the former along with the latter.

2. MODEL

This section describes our probabilistic topic model, first developing the form of its joint distribution, then sketching out the problems of inference and parameter estimation. We use the following terminology and notation throughout the rest of the paper:

1. A *note* $u \in \{A, A\sharp, B, \dots, G\sharp\}$ is the most basic unit of data. It is an element from the set of neutral pitch-classes. For easy reference, we map these pitch-classes to integer note values 0 through 11. We refer to $V = 12$ as the vocabulary size of our model.
2. A *segment* is a basic unit of time in a song (e.g., a measure). We denote the notes in the n th segment by $\mathbf{u}_n = \{u_{n1}, \dots, u_{nL}\}$, where u_{nl} is the l th note in the segment. Discarding the ordering of the notes, we can also describe each segment simply by the number of times each note occurs. We use x_n to denote the V -dimensional vector whose j th element x_n^j counts the number of times that the j th note appears in the n th segment.

3. A *song* s is a sequence of notes in N segments: $s = \{\mathbf{u}_1, \dots, \mathbf{u}_N\}$. Discarding the ordering of notes within segments, we can also describe a song by the sequence of count vectors $X = (x_1, \dots, x_N)$.
4. A music *corpus* is a collection of M songs denoted by $\mathcal{S} = \{s_1, \dots, s_M\}$.
5. A *topic* z is a probability distribution over the vocabulary of $V = 12$ pitch-classes. Topics model particular groups of notes that frequently occur together within individual segments. In practice, these groupings should contain the principle set of pitches for a particular musical key. Thus, we interpret each topic’s distribution over twelve pitch-classes as the key-profile for a musical key. We imagine that each segment in a song has its own topic (or key), and we use $\mathbf{z} = (z_1, z_2, \dots, z_N)$ to denote the sequence of topics across all segments. In western tonal music, prior knowledge suggests to look for $K = 24$ topics corresponding to the major and minor scales in each pitch-class. Section 2.3 describes how we identify the topics with these traditional key-profiles.

With this terminology, we can describe our probabilistic model for songs in a musical corpus. Note that we do not attempt to model the order of note sequences within a segment or the order of segments within a song. Just as LDA for topic modeling in text treats each document as a “bag of words”, our probabilistic model treats each song as a “bag of segments” and each segment as a “bag of notes”.

2.1 Generative process

Our approach for automatic key-profiling in music is based on the generative model of LDA for discovering topics in text. However, instead of predicting words in documents, we predict notes in songs. Our model imagines a simple, stochastic procedure in which observed notes and key-profiles are generated as random variables. In addition, we model the key-profiles as latent variables whose values must be inferred by conditioning on observed notes and using Bayes rule.

We begin by describing the process for generating a song in the corpus. First, we draw a topic weight vector that determines which topics (or keys) are likely to appear in the song. The topic weight vector is modeled as a Dirichlet random variable. Next, for each segment of the song, we sample from the topic weight vector to determine the key (e.g., A minor) of that segment. Finally, we repeatedly draw notes from the key-profile until we have generated all the notes in the segment. More formally, we can describe this generative process as follows:

1. For each song in the corpus, choose a K -dimensional topic weight vector θ from the Dirichlet distribution:

$$p(\theta|\alpha) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \prod_i \theta^{\alpha_i - 1}. \quad (1)$$

Note that α is a K -dimensional corpus-level parameter that determines which topics are likely to co-occur in individual songs. The topic weight vector satisfies $\theta_i \geq 0$ and $\sum_k \theta_k = 1$.

- For each segment indexed by $n \in \{1, \dots, N\}$ in a song, choose the topic $z_n \in \{1, 2, \dots, K\}$ from the multinomial distribution $p(z_n = k | \theta) = \theta_k$.
- For each note indexed by $\ell \in \{1, \dots, L\}$ in the n th measure, choose a pitch-class from the multinomial distribution $p(u_{n\ell} = i | z_n = j, \beta) = \beta_{ij}$. The β parameter is a $V \times K$ matrix that encodes each topic as a distribution over $V = 12$ neutral pitch-classes. Section 2.3 describes how we identify these distributions as key-profiles for particular musical keys.

This generative process specifies the joint distribution over observed and latent variables for each song in the corpus. In particular, the joint distribution is given by:

$$p(\theta, \mathbf{z}, s | \alpha, \beta) = p(\theta | \alpha) \prod_{n=1}^N p(z_n | \theta) \prod_{l=1}^{L_n} p(u_{nl} | z_n, \beta). \quad (2)$$

Figure 2(a) depicts the graphical model for the joint distribution over all songs in the corpus. As in LDA [1], we use plate notation to represent independently, identically distributed random variables within the model. Whereas LDA for text describes each document as a “bag of words”, we model each song as a “bag of segments”, and each segment as a “bag of notes”. As a result, the graphical model in Figure 2(a) contains an additional plate beyond the graphical model of LDA for text.

2.2 Inference and learning

The model in eq. (2) is fully specified by the Dirichlet parameter α and the musical key-profiles β . Suppose that these parameters are known. Then we can use probabilistic inference to analyze songs in terms of their observed notes. In particular, we can infer the main key-profile for each song as a whole, or for individual segments. Inferences are made by computing the posterior distribution

$$p(\theta, \mathbf{z} | s, \alpha, \beta) = \frac{p(\theta, \mathbf{z}, s | \alpha, \beta)}{p(s | \alpha, \beta)} \quad (3)$$

following Bayes rule. The denominator in eq. (3) is the marginal distribution, or likelihood, of a song:

$$p(s | \alpha, \beta) = \int p(\theta | \alpha) \prod_{n=1}^N \sum_{z_n=1}^K p(z_n | \theta) \prod_{l=1}^{L_n} p(u_{nl} | z_n, \beta) d\theta. \quad (4)$$

The problem of learning in our model is to choose the parameters α and β that maximize the log-likelihood of all songs in the corpus, $\mathcal{L}(\alpha, \beta) = \sum_m \log p(s_m | \alpha, \beta)$. Learning is unsupervised because we require no training set with key annotations or labels.

In latent variable models such as ours, the simplest approach to learning is maximum likelihood estimation using the Expectation-Maximization (EM) algorithm [3]. The

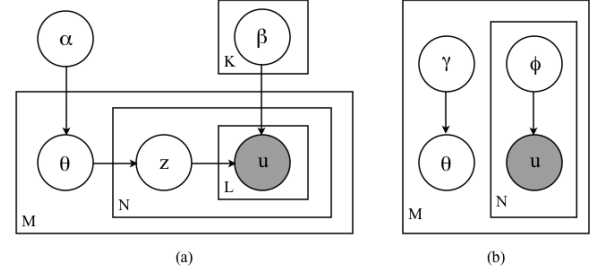


Figure 2. (a) Graphical representation of our model and (b) the variational approximation for the posterior distribution in eq. (3). See Appendix A for details.

EM algorithm iteratively updates parameters by computing expected values of the latent variables under the posterior distribution in eq. (3). In our case, the algorithm iteratively alternates between an E-step, which represents each song in the corpus as a random mixture of 24 key-profiles, and an M-step, which re-estimates the weights of the pitch classes for each key-profile. Unfortunately, these expected values cannot be analytically computed; therefore, we must resort to a strategy for approximate probabilistic inference. We have developed a variational approximation for our model based on [7] that substitutes a tractable distribution for the intractable one in eq. (3). Appendix A describes the problems of inference and learning in this approximation in more detail.

2.3 Identifying Topics as Keys

Recall from section 2.1 that the estimated parameter β expresses each topic as a distribution over $V = 12$ neutral pitch-classes. While this distribution can itself be viewed as a key-profile, an additional assumption is required to learn topics that can be identified with particular musical keys. Specifically, we assume that key-profiles for different keys are related by simple transposition: e.g., the profile for $C\sharp$ is obtained by transposing the profile for C up by one half-step. This assumption is the full extent to which our approach incorporates prior knowledge of music.

The above assumption adds a simple constraint to our learning procedure: instead of learning $V \times K$ independent elements in the β matrix, we tie diagonal elements across different keys of the same mode (major or minor). Enforcing this constraint, we find that the topic distributions learned by the EM algorithm (see section 3) can be unambiguously identified with the $K = 24$ major and minor modes of classical western music. For example, one topic distribution places its highest seven weights on the pitches $C, D, E, F, G, A,$ and B ; since these are precisely the notes of the C major scale, we can unambiguously identify this topic distribution with the key-profile for C major.

3. RESULTS

We estimated our model from a collection of 235 MIDI files compiled from <http://www.classicalmusicmidipage.com>.

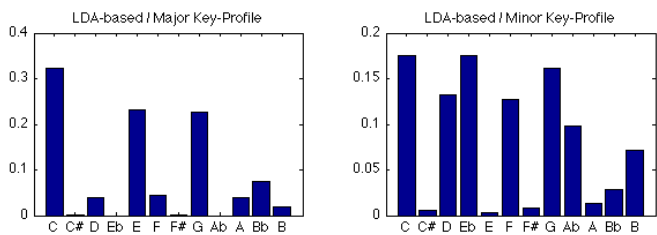


Figure 3. The C major and C minor key-profiles learned by our model, as encoded by the β matrix.

The collection included works by Bach, Vivaldi, Mozart, Beethoven, Chopin, and Rachmaninoff. These composers were chosen to span the baroque through romantic periods of western, classical music.

We experimented with different segment lengths and different ways of compiling note counts. Though measures define natural segments for music, we also experimented with half-measures and quarter-beats. All these choices led to similar musical key-profiles. We also experimented with two ways of compiling note counts within segments. The first method sets the counts proportional to the cumulative duration of notes across the segment; the second method sets the counts proportional to the number of distinct times each note is struck. We found that the second method worked best for key-finding, and we report results for this method below.

3.1 Learning Key-Profiles

Recall that each column of the estimated β matrix encodes a musical key as a distribution over $V = 12$ neutral pitch-classes. Fig. 3 shows the two columns that we identified as belonging to the keys of C major and C minor. These key-profiles have the same general shape as those of KK, though the weights for each pitch-class are not directly comparable. (In our model, these weights denote actual probabilities.) Note that in both major and minor modes, the largest weight occurs on the tonic (C), while the second and third largest weights occur on the remaining degrees of the triad (G, E for C major; G, Eb for C minor). Our key-profiles differ only in the relatively larger weight given to the minor 7th (Bb) of C major and major 7th (B) of C minor. Otherwise, the remaining degrees of the diatonic scale (D, F, A for C major; D, F, Ab for C minor) are given larger weights than the remaining chromatics. Profiles for other keys can be found by transposing.

3.2 Symbolic Key-Finding

From the posterior distribution in eq. (3), we can infer hidden variables θ and z that identify dominant keys in whole songs or segments within a song. In particular, we can identify the overall key of a song from the largest weight of the topic vector θ that maximizes eq. (3). Likewise, we can identify the key of particular segments from the most probable values of the topic latent variables z_n .

We first show results at the song-level, using our model to determine the overall key of the 235 musical pieces in

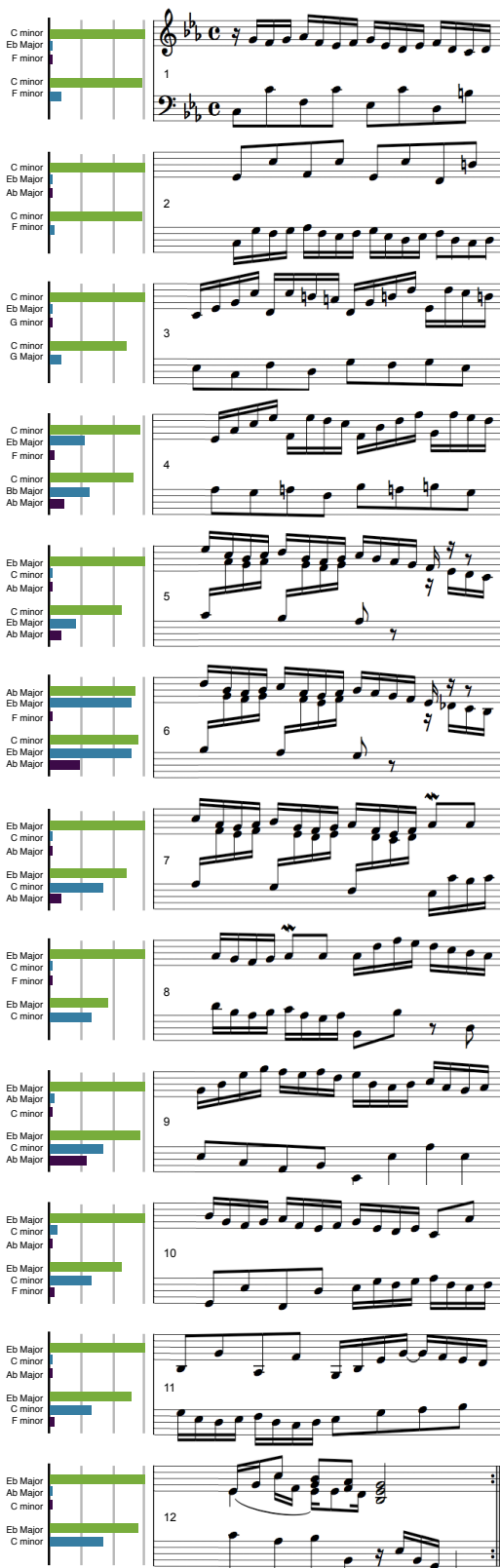


Figure 4. Key judgments for the first 12 measures of Bach’s Prelude in C minor, WTC-II. Annotations for each measure show the top three keys (and relative strengths) chosen for each measure. The top set of three annotations are judgments from our LDA-based model; the bottom set of three are from human expert judgments [8].

Song Length	All	20 beats	8 beats	4 beats
LDA	86%	77%	74%	67%
KS	80%	71%	67%	66%

Table 1. Key-finding accuracy of our LDA model and the KS model on 235 classical music pieces. Song length indicates how much of each piece was included for analysis.

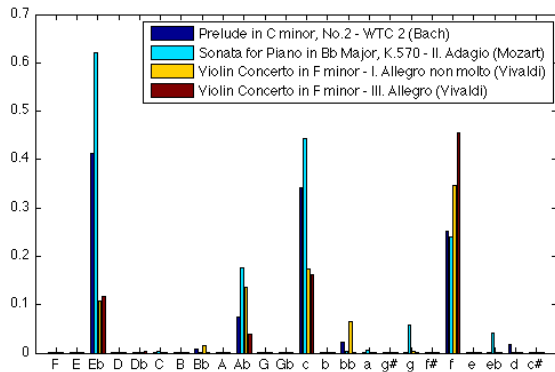


Figure 5. Songs represented as distributions over key-profiles. The first set of bars shows keys used in the query song; the remaining sets of bars show keys used in the three songs of the corpus judged to be most similar. Note how all songs modulate between the keys of Eb M, Ab M, C m, and F m.

our corpus. We tested our model against a publicly available implementation of the KS model [4] that uses normalized KK key-profiles and weighted note durations. Table 1 compares the results when various lengths of each piece are included for analysis. In this experiment, we found that our model performed better across all song lengths.

We also compared our model to three other publicly available key-finding algorithms [13]. We were only able to run these algorithms on a subset of 107 pieces in our corpus, so for these comparisons we only report results on this subset. These other algorithms used key-profiles from another implementation of the KS model [8] and from empirical analyses of key-annotated music [14, 15]. Analyzing whole songs, these other algorithms achieved accuracies between 62%–67%. Interestingly, though these models obtained their key-profiles using rule-based or supervised methods, our unsupervised model yielded significantly better results, identifying the correct key for 79% of the songs in this subset of the corpus.

Next, we show results from our model at the segment level. Fig. 4 shows how our model analyzes the first twelve measures of Bach’s Prelude in C minor from Book II of the *Well-Tempered Clavier* (WTC-II). Results are compared to annotations by a music theory expert [8]. We see that the top choice of key from our model differs from the expert judgment in only two measures (5 and 6).

3.3 Measuring Harmonic Similarity

To track key modulations within a piece, we examine its $K = 24$ topic weights. These weights indicate the propor-

tion of time that the song spends in each key. They also provide a low-dimensional description of each song’s harmonic structure. We used a symmetrized Kullback-Leibler (KL) divergence to compute a measure of dissimilarity between songs based on their topic weights. Fig. 5 shows several songs as distributions over key-profiles. (Note that previous graphs showed key-profiles as distributions over pitches.) The first set of bars show the topic weights for the same Bach Prelude analyzed in the previous section; the remaining sets of bars show the topic weights for the three songs in the corpus judged to be most similar (as measured by the symmetrized KL divergence). From the topic weight vectors, we see that all songs modulate primarily between the keys of Eb M, Ab M, C m, and F m.

4. CONCLUSION

In this paper, we have described a probabilistic model for the unsupervised learning of musical key-profiles. Unlike previous work, our approach does not require key-annotated music or make use of expert domain knowledge. Extending LDA from text to music, our model discovers latent topics that can be readily identified as the $K = 24$ primary key-profiles of western classical music. Our model can also be used to analyze songs in interesting ways: to determine the overall key, to track harmonic modulations, and to provide a low-dimensional descriptor for similarity-based ranking. Finally, though the learning in our model is unsupervised, experimental results show that it works very well compared to existing methods.

5. REFERENCES

- [1] D. M. Blei, A. Y. Ng, M.I. Jordan: “Latent Dirichlet allocation,” *Journal of Machine Learning Research*, 3:993-1022, 2003.
- [2] E. Chew: “The Spiral Array: An Algorithm for Determining Key Boundaries,” *Proc. of the Second Int. Conf. on Music and Artificial Intelligence*, 18-31, 2002.
- [3] A. Dempster, N. Laird, D. Rubin: “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society, Series B*, 39(1):1-38, 1977.
- [4] T. Eerola, P. Toiviainen: “MIDI Toolbox: MATLAB Tools for Music Research,” *University of Jyväskylä, Jyväskylä, Finland*, Available: <http://www.jyu.fi/musica/miditoolbox/>, 2004.
- [5] L. Fei-Fei, P. Perona: “A Bayesian hierarchical model for learning natural scene categories,” *CVPR*, 524-531, 2005.
- [6] T. Griffiths, M. Steyvers, D. Blei, J. Tenenbaum: “Integrating topics and syntax,” In L. Saul, Y. Weiss, and L. Bottou, editors, *NIPS*, 537-544, 2005.
- [7] M. I. Jordan, Z. Ghahramani, T. Jaakkola, L. Saul: “Introduction to variational methods for graphical models,” *Machine Learning*, 37:183-233, 1999.

- [8] C. Krumhansl: *Cognitive Foundations of Musical Pitch*, Oxford University Press, Oxford, 1990.
- [9] C. Krumhansl, E. J. Kessler: "Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys," *Psychological Review*, 89:334-68, 1982.
- [10] H. C. Longuet-Higgins, M. J. Steedman: "On interpreting Bach," *Machine Intelligence*, 6:221-41, 1971.
- [11] B. Marlin: "Modeling user rating profiles for collaborative filtering," In S. Thrun and L. Saul and B. Schölkopf, editors, *NIPS*, 2003.
- [12] D. Rizo: "Tree model of symbolic music for tonality guessing," *Proc. of the Int. Conf. on Artificial Intelligence and Applications*, 299-304, 2006.
- [13] D. Sleator, D. Temperley: "The Melisma Music Analyzer," Available: <http://www.link.cs.cmu.edu/music-analysis/>, 2001.
- [14] D. Temperley: *The Cognition of Basic Musical Structure*, MIT Press, 2001.
- [15] D. Temperley: "A Bayesian approach to key-finding," *Lecture Notes in Computer Science*, 2445:195-206, 2002.

A. VARIATIONAL APPROXIMATION

This appendix describes our variational approximation for inference and learning mentioned in section 2. It is similar to the approximation originally developed for LDA [1].

A.1 Variational Inference

The variational approximation for our model substitutes a tractable distribution for the intractable posterior distribution that appears in eq. (3). At a high level, the approximation consists of two steps. First, we constrain the tractable distribution to belong to a parameterized family of distributions whose statistics are easy to compute. Next, we attempt to select the particular member of this family that best approximates the true posterior distribution.

Figure 2(b) illustrates the graphical model for the approximating family of tractable distributions. The tractable model $q(\theta, \mathbf{z}|\gamma, \phi)$ drops edges that make the original model intractable. It has the simple, factorial form:

$$q(\theta, \mathbf{z}|\gamma, \phi) = q(\theta|\gamma) \prod_{n=1}^N q(z_n|\phi_n) \quad (5)$$

We assume that the distribution $q(\theta|\gamma)$ is Dirichlet with variational parameter γ , while the distributions $q(z_n|\phi_n)$ are multinomial with variational parameters ϕ_n . For each song, we seek a factorial distribution of the form in eq. (5) to approximate the true posterior distribution in eq. (3). Or more specifically, for each song s_m , we seek the variational parameters γ_m and ϕ_m such that $q(\theta, \mathbf{z}|\gamma_m, \phi_m)$ best matches $p(\theta, \mathbf{z}|s_m, \alpha, \beta)$.

Though it is intractable to compute the statistics of the true posterior distribution $p(\theta, \mathbf{z}|\alpha, \beta)$ in eq. (3), it is possible to compute the Kullback-Leibler (KL) divergence

$$\text{KL}(q, p) = \sum_{\mathbf{z}} \int d\theta q(\theta, \mathbf{z}|\gamma, \phi) \log \frac{q(\theta, \mathbf{z}|\gamma, \phi)}{p(\theta, \mathbf{z}|s, \alpha, \beta)} \quad (6)$$

up to a constant term that does not depend on γ and ϕ . Note that the KL divergence measures the quality of the variational approximation. Thus, the best approximation is obtained by minimizing the KL divergence in eq. (6) with respect to the variational parameters γ and ϕ_n . To derive update rules for these parameters, we simply differentiate the KL divergence and set its partial derivatives equal to zero. The update rule for γ_m is analogous to the one in the LDA model for text documents [1]. The update rule for the multinomial parameters ϕ_{ni} is given by:

$$\phi_{ni} \propto \prod_{j=1}^V \beta_{ij}^{x_n^j} \exp[\Psi(\gamma_i)], \quad (7)$$

where $\Psi(\cdot)$ denotes the digamma function and x_n^j denotes the count of the j th pitch class in the n th segment of the song. We omit the details of this derivation, but refer the reader to the original work on LDA [1] for more detail.

A.2 Variational Learning

The variational approximation in eq. (5) can also be used to derive a lower bound on the log-likelihood $\log p(s|\alpha, \beta)$ of a song s . Summing these lower bounds over all songs in the corpus, we obtain a lower bound $\ell(\alpha, \beta, \gamma, \phi)$ on the total log-likelihood $\mathcal{L}(\alpha, \beta) = \sum_m \log p(s_m|\alpha, \beta)$. Note that the bound $\ell(\alpha, \beta, \gamma, \phi) \leq \mathcal{L}(\alpha, \beta)$ depends on the model parameters α and β as well as the variational parameters γ and ϕ across all songs in the corpus.

The variational EM algorithm for our model estimates the parameters α and β to maximize this lower bound. It alternates between two steps:

1. (E-step) Fix the current model parameters α and β , compute variational parameters $\{\gamma_m, \phi_m\}$ for each song s_m by minimizing the KL divergence in eq. (6).
2. (M-step) Fix the current variational parameters γ and ϕ across all songs from the E-step, maximize the lower bound $\ell(\alpha, \beta, \gamma, \phi)$ with respect to α and β .

These two steps are repeated until the lower bound on the log likelihood converges to a desired accuracy. The updates for α and β in the M-step are straightforward to derive. The update rule for β is given by:

$$\beta_{ij} \propto \sum_{m=1}^M \sum_{n=1}^N \phi_{mn}^i x_{mn}^j. \quad (8)$$

While the count x_{mn}^j in eq. (8) may be greater than one, this update is otherwise identical to its counterpart in the LDA model for text documents. The update rule for α also has the same form.