Structured Prediction Models for Chord Transcription of Music Audio

Adrian Weller, Daniel Ellis, Tony Jebara
Columbia University, New York, NY 10027
aw2506@columbia.edu, dpwe@ee.columbia.edu, jebara@cs.columbia.edu

Abstract

Chord sequences are a compact and useful description of music, representing each beat or measure in terms of a likely distribution over individual notes without specifying the notes exactly. Transcribing music audio into chord sequences is essential for harmonic analysis, and would be an important component in content-based retrieval and indexing, but accuracy rates remain fairly low. In this paper, the existing 2008 LabROSA Supervised Chord Recognition System is modified by using different machine learning methods for decoding structural information, thereby achieving significantly superior results. Specifically, the hidden Markov model is replaced by a large margin structured prediction approach (SVMstruct) using an enlarged feature space. Performance is significantly improved by incorporating features from future (but not past) frames. The benefit of SVMstruct increases with the size of the training set, as might be expected when comparing discriminative and generative models. Without yet exploring non-linear kernels, these improvements lead to state-of-the-art performance in chord transcription. The techniques could prove useful in other sequential learning tasks which currently employ HMMs.

1. Introduction

Humans are able to extract rich and meaningful information from complex audio performances, but so far it has proved difficult for computers to deal with these signals, particularly when attempting a challenging task such as transcribing chords from ensemble performances of popular music.

This paper is motivated by the Music Information Retrieval Evaluation eXchange (MIREX) [1], a contest where entrants were judged on how well they were able to identify the chords in commercial recordings of popular music. The evaluation was performed over a set of manually-labeled Beatles songs [2]. Chord labels were simplified to 25 possibilities – one for each of the 12 major chords, one for each of the 12 minor chords, and one additional label to represent 'no chord'. The top performing methods included the LabROSA Supervised Chord Recognition System [3] which obtained the second highest accuracy in the evaluation, scoring about 10% relative worse than the best system [4], [20]. The modifications reported in this paper have improved the LabROSA system performance by approximately an 8% relative increase, which is almost equivalent to the state-of-the-art. By appealing to a well-established large margin discriminative methodology that has been popularized by support vector machines, this performance is achieved without extensive tweaking or domain adaptation. [20] discusses a technique used first to suppress drum sounds from an audio input in order to obtain a harmonic-emphasized signal, which is then processed in a similar manner to the LabROSA system, i.e. by extracting chroma features and decoding using an HMM as in [17]. While this pre-processing technique is not incorporated here, one of the attractive aspects of our approach is that it may be combined with other such ideas, and also may be augmented by adding additional features, without significant reformulation of the underlying algorithms. Further increases in performance are also possible through the investigation of more elaborate nonlinear kernels.

The main stages of the LabROSA system may be summarized thus: An input song is first converted into beat-synchronous frames (for the Beatles songs used, the average number of frames per song is 459, with a range of 77 to 1806), each with 12 chroma features which are constructed to estimate the intensity of each semitone regardless of octave. Each of these 12 features is in the range [0,1]. It is assumed that the chord is constant within a frame. These processes are described in [9]. The remaining task is then a sequence labeling problem where each frame is treated as one token. For the MIREX contest, the frame labels are converted to a sequence of chords with the times of the changes. For purposes of this paper, this last conversion is not performed but instead accuracy per frame is used as the metric.

The sequence labeling method employed by the LabROSA system is a Hidden Markov Model with Gaussian emissions. From the training data, a single full-covariance Gaussian distribution is fit to all major chord instances, each rotated by the appropriate number of semitones to "transpose" them to a common root note.

Similarly a minor-chord Gaussian is fit. These are then rotated through each semitone to provide models of the emission probabilities for each chord state. The transition matrix is estimated by counting the transitions in the training data, along with a small prior (also known as Laplace smoothing) to avoid zero transition probabilities in the matrix.

2. Hidden Markov Models

A Hidden Markov Model (HMM) is a statistical model where the underlying process is assumed to be Markov. The state itself is unobservable or hidden; instead each state has a probability distribution over a range of possible emissions or output tokens, which are observed. After its parameters have been trained on labeled data, an HMM can be used on unlabeled data to take an observation sequence as input and infer or decode the likely states that generated the sequence.

HMMs have a long history and have been successfully applied to various tasks involving labeling sequential information. Among other applications, they have been used for speech recognition [15] and bioinformatics [7]. Since both speech and chord recognition involve the sequential decoding of audio signals, it was natural to consider this approach for chord transcription [17].

The baseline LabROSA system uses Viterbi decoding to compute the most likely sequence of chord labels from a song's chroma features. It is also possible to compute the most likely label for a sequence on a token by token basis (following the terminology of [15], call this 'MaxGamma' decoding). Since the evaluation metric considers period by period accuracy, we investigated the impact on performance of using MaxGamma decoding, rather than Viterbi. As presented in section 7 below, this generally provided a slight improvement.

3. Generative and Discriminative Models

While HMMs are a natural choice for the model, the estimation approach for fitting HMMs to data is typically maximum likelihood, a so-called generative criterion. Rather than focus the resources of the model on the input-output task required, generative approaches merely fit the model to data without any task specificity. Discriminative methods, conversely, estimate model parameters to achieve accurate input-output mappings and are more directly relevant for the sequence labeling problem [10]. A discriminative contender to the maximum likelihood hidden Markov model is the conditional random field (CRF) [12] which maximizes the conditional likelihood $p(y|x)$ of a label sequence y given an input sequence of features x. There exist more aggressive schemes involving maximum margin structured prediction which focus directly on the mapping from input features to output labels, potentially achieving further gains by avoiding explicit density modeling. As popularized in [21], "one should solve the [classification] problem directly and never solve a more general problem as an intermediate step." For chord transcription, the goal is simply to map feature observations to chord label states $x \rightarrow y$. An approach which does this directly (such as SVMstruct, described in section 4 below) would be expected to provide the greatest accuracy.

There are, however, arguments to support a generative strategy. One reason is that if the assumptions made in the model are good – in this application, if Gaussian emission probabilities are a good description of musical feature data – then it is expected that such a model will perform better than a discriminative one which uses less prior knowledge about the system, particularly when the amount of training data is small. Similar arguments for the use of generative modeling were presented in [13] and the behaviors of generative and discriminative approaches are explored in the experiment results in section 7 below.

Other reasons to favor a strategy which models the observation distributions include the ability to output measures such as the confidence of a prediction, or the second most likely label for a token. These are not required for the task here, but could be important elements of other systems.

4. Large Margin Structured Prediction

Maximum margin classifiers have been in use for many years and have also accommodated the use of soft margins via slack variables [6]. These allow classification hyperplanes such as the Support Vector Machine (SVM) to find an optimal split even when training data is non-separable. A tuning parameter C allows the user to control the degree to which the algorithm trades off margin breadth against training error. Recently the maximum margin approach has been extended to structured classification problems such as sequence labeling [18], [19]. Although this framework generally involves an intractably large (exponential) number of constraints, the SVMstruct algorithm uses a cutting plane approach that can provide a solution within polynomial time.

The goal is to estimate a function $f : X \rightarrow Y$, from the input space of features, X, to a discrete output space of labels, Y. This is accomplished by finding a suitable augmented function $F : X \times Y \rightarrow \boldsymbol{R}$ with parameter vector $\boldsymbol{w}$ such that

$$f(\boldsymbol{x}) = \arg\max_{y \in Y} F(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{w})$$

F is taken to be linear so that $F(\boldsymbol{x},\boldsymbol{y};\boldsymbol{w}) = \langle \boldsymbol{\psi}(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{w} \rangle$, with $\boldsymbol{\psi}(\boldsymbol{x},\boldsymbol{y})$ a joint feature map which depends on the application.

Suppose the training set is $(\boldsymbol{x}_i,\boldsymbol{y}_i)$ for $i \in N = \{1..n\}$. If the data is separable (that is, if the model can achieve perfect accuracy on the training data), then in order to pick out the right $\boldsymbol{y}_i$ for a given $\boldsymbol{x}_i$, the following constraint is imposed:

$$\langle \boldsymbol{\psi}(\boldsymbol{x}_i, \boldsymbol{y}_i), \boldsymbol{w}\rangle - \langle \boldsymbol{\psi}(\boldsymbol{x}_i, \boldsymbol{y}), \boldsymbol{w}\rangle \geq 1, \forall \boldsymbol{y} \neq \boldsymbol{y}_i$$

As in a standard SVM, $\boldsymbol{w}$ is found which maximizes the margin by minimizing $\|\boldsymbol{w}\|$ subject to the above constraint. Thus in typical notation the problem is

$$\min_{\boldsymbol{w}} \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{w}\rangle \text{ s.t.} \langle \boldsymbol{\psi}(\boldsymbol{x}_i, \boldsymbol{y}_i), \boldsymbol{w}\rangle - \langle \boldsymbol{\psi}(\boldsymbol{x}_i, \boldsymbol{y}), \boldsymbol{w}\rangle \geq 1,$$
$$\forall i \in N, \boldsymbol{y} \neq \boldsymbol{y}_i$$

When the data is non-separable (as in this application), slack variables ξ_{i} are introduced. As is customary, writing $\boldsymbol{\delta\psi}_{\mathrm{i}}(\boldsymbol{y}) = \boldsymbol{\psi}(\boldsymbol{x}_i, \boldsymbol{y}_i) - \boldsymbol{\psi}(\boldsymbol{x}_i, \boldsymbol{y})$, the SVMstruct formulation is:

$$\min_{\boldsymbol{w},\xi} \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{w}\rangle + \frac{C}{n}\sum_{i=1}^{n} \xi_{\mathrm{i}} \text{ s.t. } \xi_{\mathrm{i}} \geq 0 \ \forall i \in N,$$
$$\langle \boldsymbol{\delta\psi}_{\mathrm{i}}(\boldsymbol{y}), \boldsymbol{w}\rangle \geq 1 - \xi_{\mathrm{i}}, \forall i \in N, \boldsymbol{y} \neq \boldsymbol{y}_i$$

where as above $C > 0$ is a tuning parameter controlling the tradeoff between the margin breadth and the slack variables.

5. Model selection

Both CRFs and SVMstruct were included in a survey of models for structured learning problems [14], which compared performance on part-of-speech tagging (POS) and Optical Character Recognition (OCR) tasks. The conclusion was that SVMstruct provides the best performance on both tasks. This result has been challenged - [16] suggests that if appropriate adjustments are made to the software used in [14] in order to ensure that the different approaches use identical feature functions, then the CRF and SVMstruct approaches have similar peak results – but still the consensus is that SVMstruct typically performs at least as well as other methods.

There are two further advantages of using SVMstruct for chord transcription compared to the baseline HMM approach. First, SVMstruct has better regularization properties and reduces the risk of over-fitting to the training data particularly when many features are added to the input sequence. Noting this, the effect of adding features from adjacent frames, as suggested in [5], is explored in section 7 below. Secondly, it opens up the large toolbox of kernel methods, which could prove fruitful in future work but is not fully explored here.

[14] also introduced a Structured Learning Ensemble (SLE) method for combining the power of several different structural learning models. The SLE method performed slightly better than the others in [14], and may provide another productive area for further work.

6. Experiments

All experiments were performed on frame-level data, using the 25 possible chord labels described in section 1. Various models were compared after having been trained and tested on data sets chosen from the universe of 180 labeled Beatles songs.

Specifically, ten random permutations of all songs were selected. For each permutation, every model was trained on the first train% (30%, 60% or 90%) of the 180 permuted songs. The last 10% of the permuted songs was used for testing, and for validation if required, irrespective of the amount of training data used. Since the HMM models do not require a validation set, they were simply tested on the entire final 10% of the songs. The SVMstruct models, however, require the estimation of a *C* parameter. This was achieved by splitting the final 10% into two halves – the penultimate 5% of the permuted songs, call this set A, and the last 5%, call this set B. Each model was trained with a broad range of values of *C*. The particular value which gave optimal performance on set A was used for testing the model on set B, and vice versa. The results on A and B were then combined by averaging, weighted by the respective number of frames, to give the accuracy per frame over the entire test set. This approach meant that for each permutation of the songs, as train% varied, all models used the same test data, facilitating comparison.

For the SVMstruct runs, code from [11] was used. The specific instantiation was SVM-HMM with the following parameters: the precision constant e (epsilon) set to 0.1, the order of dependencies of transitions in HMM t set to the default 1, and the order of dependencies of emissions in HMM e set to the default 0. With these settings, the interdependency structure of the features and labels in the model is comparable to that of the HMM used in the LabROSA system [3]. The order of dependency of transitions was changed from 1 to 2 on a few test runs to investigate its impact. These runs took significantly longer and did not improve accuracy so are not reported here, but may warrant future investigation.

The LabROSA system's HMM uses Gaussian emissions, which lead to curved (quadratic) decision boundaries between labels. Since in this paper only linear kernels for SVMstruct are considered, to allow comparison against the HMM results and as a first step towards more sophisticated kernels, in some runs quadratic terms were added, i.e. pairwise products of existing features were added as new features. Features from neighboring frames were also introduced in some models, as discussed in section 5 above.

7. Results

To judge results, Hamming distance was used, that is the label predicted for each frame was either correct or not. Frame accuracy was not computed per song and then combined to give each song equal weight, but rather frame accuracy was calculated over the entire test set – which had a different number of frames for each permutation of the songs. This relates to the measure used in the MIREX 2008 contest, and is the metric used for training all the models.

Figure 1 displays the Hamming accuracy for each structured prediction model as the amount of training data was varied, where results have been averaged over the ten random permutations of songs used. Note results shown are lower than those reported in MIREX 2008 due to differences in ground truth alignment, but this does not affect comparisons between models.

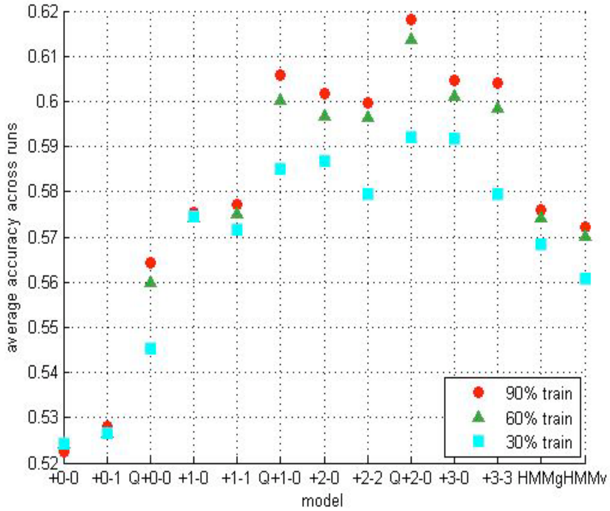

Figure 1. Average Hamming accuracies for each model

The model on the far right of the figure, *HMMv*, is the baseline HMM approach with Viterbi decoding that was used in the LabROSA system. To its left, *HMMg* is the same HMM model but using MaxGamma decoding as described in section 2, showing a small improvement.

The models to the left are all SVMstruct runs using various feature combinations. *+0-0* on the far left uses only the original 12 chroma features for each frame – the same features used by the HMM models. To its right, *+0-1* adds the features of the previous frame, so in this model each frame is represented by 24 dimensions. Next to the right, *Q+0-0* uses the 12 original features and adds all quadratic cross-terms for a total of 90 features. To its right, *+1-0* uses the current frame's 12 features and adds the next frame's 12 features. Next to the right, *+1-1* uses the current frame's features along with those from both one frame forward and one frame backward.

Further to the right, the remaining models add more features and are similarly labeled: *+m-n* uses features from the current frame along with those from each of the next *m* frames and each of the previous *n* frames; *Q* at the front means that in addition, all quadratic cross-terms have been added. There's one exception, *Q+2-0*, which does not use all cross-terms since that would have led to an unwieldy 702 dimensions, but instead uses the 324 features from *Q+1-0* and adds just the 12 additional chroma features from 2 frames ahead without then adding more cross-terms.

Results are almost uniformly better as the size of the training set grows, with the rates of improvement of the more complex SVMstruct models higher than those of the HMMs. Based on the modest gains from going from 60% to 90% training set size, however, there may not be much more to gain with more training data. Quadratic terms provide dramatic improvements, suggesting further gains may be achieved with non-linear kernels. Adding features from future frames also provides striking benefits, but only up to two frames ahead. Adding features from past frames appears not to help.

Although the SVMstruct approach showed substantial improvements when certain additional features were added, incorporating those same features into the HMM framework led to worse performance (results not shown here).

Table 1. Average accuracies and std deviations

	30% trained	60% trained	90% trained
+0-0	0.524 ± 0.062	0.524 ± 0.060	0.523 ± 0.058
+0-1	0.527 ± 0.064	0.527 ± 0.060	0.528 ± 0.059
Q+0-0	0.545 ± 0.053	0.560 ± 0.054	0.564 ± 0.054
+1-0	0.575 ± 0.059	0.574 ± 0.056	0.575 ± 0.060
+1-1	0.572 ± 0.061	0.575 ± 0.058	0.577 ± 0.058
Q+1-0	0.585 ± 0.058	0.600 ± 0.056	0.606 ± 0.054
+2-0	0.587 ± 0.059	0.597 ± 0.055	0.602 ± 0.056
+2-2	0.580 ± 0.058	0.597 ± 0.056	0.600 ± 0.058
Q+2-0	0.592 ± 0.056	0.614 ± 0.053	0.618 ± 0.052
+3-0	0.592 ± 0.055	0.601 ± 0.050	0.605 ± 0.051
+3-3	0.580 ± 0.055	0.599 ± 0.051	0.604 ± 0.050
HMMg	0.568 ± 0.053	0.574 ± 0.050	0.576 ± 0.049
HMMv	0.561 ± 0.051	0.570 ± 0.047	0.572 ± 0.045

Table 1 shows the same average accuracies, along with sample standard deviations. The deviations do not differ greatly by model or as the amount of training data is varied. Although the deviations of each model individually are high relative to the observed differences in performance, because all models were trained and tested on the same data sets, relative performance can be examined using paired t-tests to obtain significant results.

Table 2. p-values for outperformance vs. HMMv

	30% trained	60% trained	90% trained
+0-0	0.998	0.999	1.000
+0-1	0.995	0.998	1.000
Q+0-0	0.985	0.945	0.897
+1-0	0.091	0.352	0.392
+1-1	0.151	0.303	0.329
Q+1-0	***0.013***	***0.001***	***0.001***
+2-0	***0.017***	***0.004***	***0.006***
+2-2	0.052	***0.004***	***0.009***
Q+2-0	***0.002***	***0.000***	***0.000***
+3-0	***0.009***	***0.001***	***0.001***
+3-3	0.060	***0.001***	***0.001***
HMMg	***0.012***	***0.008***	***0.048***

Table 2 shows p-values for paired t-tests examining outperformance of each model compared to the baseline *HMMv* approach. Small values indicate statistical significance. p-values indicating outperformance at the 5% significance level have been marked. The first five SVMstruct models do not show significant superior accuracy, but all the others do. In addition, *HMMg* shows statistically significant performance advantages over *HMMv*, although from the average accuracies it can be seen that the magnitude of this outperformance is small.

8. Alternative metric

Although accuracy in the MIREX contest was judged using Hamming distance, where each chord was determined to be either correct or incorrect and then accordingly received a score of either 1 or 0, it is reasonable to suggest that for this task all errors are not equal – e.g. C major and C# major are musically very far apart, whereas C major and A minor are closely related. The A minor root triad of A-C-E shares two notes with C major's root triad of C-E-G, which is described musically as its relative major. One suggestion for an alternative metric follows: assign a similarity score of 1/3 for each note the automatically labeled chord's root triad has in common with the true chord, so each predicted label can score either 0, 1/3, 2/3 or 1. C to C# would have a similarity score of 0; C to A minor would have a similarity score of 2/3. This triad overlap metric (TOM) is simple and is closer to human musical interpretation.

Implementing a scheme like TOM to test a model is simple. To incorporate it into the training of a model is, however, more challenging and would require rewriting of the underlying algorithm. Figure 2 shows the TOM test accuracies for all the models *trained using Hamming distance* as before. Note that by definition, always TOM ≥ Hamming distance leading to higher accuracy scores.

The results are qualitatively very similar to the Hamming accuracies shown in section 7, and the p-values (not shown) are also similar with all the same models outperforming HMMv with statistical significance. These results are somewhat remarkable when viewed in context. As discussed in section 3, the SVMstruct approach is designed to focus directly on minimizing a (regularized) training error so as to achieve a corresponding low error when testing *with the same measure*. Training so as to minimize Hamming distance and then testing with the more musical TOM is in some way evaluating the intrinsic musicality of the model – which we might expect to be higher for the generative HMM model which has been shown to fit well to music audio than for a discriminative model honed exclusively to perform well on the training metric.

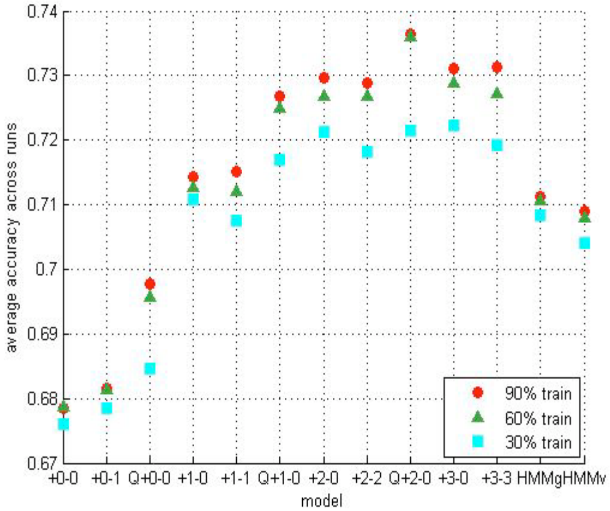

Figure 2. Average TOM accuracies for each model trained using Hamming distance

9. Conclusion and Future Work

Music is typically a highly structured art form, using certain chord progressions frequently (e.g. I-IV-V-I). It is natural, therefore, to try structured prediction models for automated chord transcription. Following existing approaches which use HMMs, the application of a more recent machine learning algorithm, SVMstruct, was employed to decode the structural information.

One of the benefits of this discriminative approach is the ability to add many features to see if they improve performance, with less risk of over-fitting than with the generative HMM model. The form of SVMstruct that was used automatically incorporates interdependency between adjacent chord labels (as in an HMM). It is interesting that adding the full feature set of forward frames gave rise to significant performance improvement, up to a saturation point. This may not be surprising since having more information about where one is heading naturally may provide useful information about where one is now. It seems remarkable, however, that similar information about previous frames does not help accuracy. This may be relevant for other musical analysis.

By using features from just the next two frames forward and some quadratic terms, SVMstruct demonstrates dramatic improvements over HMM, achieving results in the realm of state-of-the-art. There remain promising avenues to explore which may yield further improvements.

Non-linear kernels may prove helpful, as they have in other applications. Combining different models into an ensemble approach is another interesting area for further work. The LabROSA approach fits just one major and one minor chord model, rotating them through each semitone to provide a complete set of models. Neither this idea, nor any other musical knowledge, was used in the SVMstruct approach here. One way to accomplish something similar would be to take all the training data and transpose or rotate it through each semitone to yield a twelve-fold increase in training data size, which would require greater time and memory resources but should yield some performance improvement.

The approaches described here may be helpful for other audio processing sequence labeling tasks, including melody or bass line transcription and perhaps speech recognition.

10. Acknowledgments

The authors thank J. Jensen for usage of the ISP Toolbox and its optimized, MEX-based chroma calculations. The authors thank K. Murphy for usage of the HMM Toolbox for Viterbi and Gamma decoding. The authors thank C. Harte for providing Beatles song transcripts. The authors thank T. Joachims for usage of the SVMstruct code.

11. References

[1] http://www.music-ir.org/mirex/2008
[2] http://mir-research.blogspot.com/2007/09/beatles-chord-transcriptions.html
[3] http://labrosa.ee.columbia.edu/projects/chords/
[4] http://www.music-ir.org/mirex/2008/index.php/Audio_Chord_Detection_Results
[5] Y. Altun, I. Tsochantaridis and T. Hofmann, "Hidden Markov Support Vector Machines", *ICML* 2003
[6] C. Cortes, and V. Vapnik, "Support vector networks", *Machine Learning*, 20:273-297, 1995
[7] R. Durbin, S.R. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press, 1998
[8] D.P.W. Ellis, "Beat Tracking with Dynamic Programming", *MIREX* 2006
[9] D.P.W. Ellis, and G.E. Poliner, "Identifying Cover Songs with Chroma Features and Dynamic Programming Beat Tracking", *Proceedings of ICASSP* 2007:IV-1429-1432
[10] T. Jebara, *Machine Learning: Discriminative and Generative*, Kluwer, 2003. ISBN 1-4020-7647-9.
[11] T. Joachims, SVMstruct code at http://svmlight.joachims.org/svm_struct.html
[12] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data", *ICML* 2001
[13] A. Ng, and M. Jordan, "On Discriminative vs. Generative classifiers: A comparison of logistic regression and naïve Bayes", *NIPS* 2001
[14] N. Nguyen and Y. Guo, "Comparisons of Sequence Labeling Algorithms and Extensions", *ICML* 2007
[15] L. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition", *Proceedings of the IEEE* 77(2):257-286
[16] S. Sathiya Keerthi and S. Sundararajan, "CRF versus SVM-Struct for Sequence Labeling", *Yahoo Research Technical Report* 2009
[17] A. Sheh and D.P.W. Ellis, "Chord Segmentation and Recognition using EM-Trained Hidden Markov Models", *ISMIR* 2003
[18] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support Vector Machine Learning for Interdependent and Structured Output Spaces", *ICML* 2004
[19] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large Margin Methods for Structured and Interdependent Output Variables", *The Journal of Machine Learning Research,*Vol 6, December 2005:1453-1484
[20] Y. Uchiyama, K. Miyamoto, N. Ono, and S. Sagayama, "Automated Chord Detection using Harmonic Sound Emphasized Chroma from Musical Acoustic Signal", at http://www.music-ir.org/mirex/2008/abs/uchiyamamirex2008.pdf
[21] V.N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, 1998