The dangers of parsimony in query-by-humming applications

Colin Meek
University of Michigan
1101 Beal Avenue
Ann Arbor MI 48109 USA
meek@umich.edu

William P. Birmingham
University of Michigan
1101 Beal Avenue
Ann Arbor MI 48109 USA
wpb@umich.edu

Abstract

Query-by-humming systems attempt to address the needs of the non-expert user, for whom the most natural query format – for the purposes of finding a tune, hook or melody of unknown providence – is to sing it. While human listeners are quite tolerant of error in these queries, a music retrieval mechanism must explicitly model such errors in order to perform its task. We will present a unifying view of existing models, illuminating the assumptions underlying their respective designs, and demonstrating where such assumptions succeed and fail, through analysis and real-world experiments.

1 Introduction

When auditing a sung query – or indeed any musical production – a trained ear can recognize certain problems: pitch drift, out of tune notes, rhythm errors, unsteady tempo, and so forth. It is quite natural for a music teacher to comment to a student that "the third note was flat", or "you're speeding up in the third measure". These two statements represent two fundamentally different views of error: the first indicates a belief that a single note was "off", and the second indicates a belief that a trend is occurring.

The two views are, however, reconcilable. The teacher could also, at the expense of clarity, assert that "you modulated down to $F\#$ major on the third note, and modulated back to G major on the fourth note" or "all of the notes after the third measure were too short" respectively. Thus, it may seem reasonable in the context of a query-by-humming (QBH) system to view errors in one of two fundamental ways:

- Error occurs locally: any discrepancy between a query and its target must be explained on a note by note (or frame by frame) basis, though allowing for some overall differences in register, key and tempo. This view reasonably models the situation described by the first statement (the note is flat), and accounts for the other situation indirectly (these notes are all too short).
- An error is always "cumulative": errors occur with respect to the context established by previous notes. This view reasonably models the second statement (the tempo increases), and accounts for the other situation indirectly (seen as a modulation down, then a modulation up).

With respect to pitch and rhythm, most existing QBH systems implicitly make one or the other assumption. There are compelling arguments in favor of such assumptions, particularly with regards to model complexity. In addition, neither assumption is fatal even if both types of error are prevalent, as the alternate interpretations shown above reveal.

Why is model parsimony dangerous? As the size of a database increases, intelligently diagnosing error becomes more and more critical: if we can explain a query with respect to its target in terms of one error rather than four, the group of songs that appear "just as close" is much smaller. Of course, most models do not simply count the number of errors in a match, but the intuition remains the same. In Section 5, we formalize a more general form of this observation.

2 Errors

A query model should be capable of expressing the following musical – or un-musical you might argue – transformations, relative to a target:

1. Insertions and deletions: adding or removing notes from the target, respectively. These *edits* are frequently introduced by transcription tools as well.
2. Transposition: the query may be sung in a different *key* or *register* than the target. Essentially, the query might sound "higher" or "lower" than the target.
3. Tempo: the query may be slower or faster than the target.
4. Modulation: over the course of a query, the transposition may change.
5. Tempo change: the singer may speed up or slow down during a query.
6. Non-cumulative local error: the singer might sing a note off-pitch or with poor rhythm.

2.1 Edit Errors

Insertions and deletions in music tend to influence surrounding events (Mongeau and Sankoff, 1990). For instance, when an insertion is made, the inserted event and its neighbor tend to occupy the temporal space of the original note: if an insertion is made and the duration of the neighbors is not modified, the underlying rhythmic structure (the beat) is changed. Similarly, insertions will tend to modify the intervallic contour of a phrase, to maintain the overall contour. Reflecting this process, we describe the edit operations as "elaborations" and "joins" for insertion and deletion respectively, because the inserted notes are seen as embellishing the original parent note, and deleted notes result in the merging of multiple notes into a longer one.

This approach to edits reflects a natural musical interpretation. A pragmatic motivation for our "musical" definition of edit is transcriber error. In this context, we clearly would not expect the onset times or pitches of surrounding events to be influenced by a "false hit" insertion, or a missed note. The relationships amongst successive events must therefore be modified to avoid warping and modulation. Reflecting this bias, we use the terms "join" and "elaboration" to refer to deletions and insertions, respectively.

2.2 Transposition and Tempo

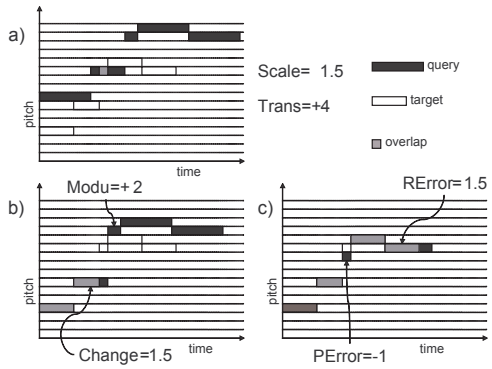

Figure 1: Examples of errors

We account for the phenomenon of persons reproducing the same "tune" at different speeds and in different registers or keys. Few people have the ability to remember and reproduce exact pitches (Terhardt and Ward, 1982), an ability known as "absolute" or "perfect" pitch. As such, transpositional invariance is a desirable feature of any query/retrieval model. The effect of transposition is simply to add a certain value to all pitches. Consider for example the transposition illustrated in Figure 1, Section a, of $Trans$ = +4.

Tempo in this context is simply the translation of rhythm, which describes duration relationships, into actual time durations. Again, it is difficult to remember and reproduce an exact tempo. Moreover, it is very unlikely that two persons would choose the same metronome marking, much less unconstrained beat timing, for any piece of music. This is a natural "musical" interpretation. The effect of a tempo scaling is simply to multiply all inter-onset interval (IOI) values by some amount, where the IOI is the time between the onsets of successive notes. Thus, if the query is 50% slower than the target, we have a scaling value of $Tempo = 1.5$, as shown in Figure 1, Section a.

2.3 Modulation and tempo change

Throughout a query, the degree of transposition or tempo scaling can change, referred to as *modulation* and *tempo change*, respectively. Consider a query beginning with the identity transposition $Trans = 0$ and identity tempo scaling $Tempo = 1$, as in Figure 1, Section b. When a modulation or tempo change is introduced, it is always with respect to the previous transposition and tempo. For instance, on the third note of the example, a modulation of $Modu = +2$ occurs. For the remainder of the query, the transposition is equal to $0 + 2 = +2$, from the starting reference transposition of 0. Similarly, the tempo change of $Change = 1.5$ on the second note means that all subsequent events occur at a tempo scaling of $1 \cdot 1.5 = 1.5$.

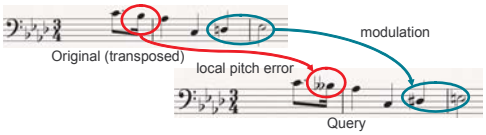

Figure 2: Portion of a query on the *American National Anthem*

2.4 Local Pitch and IOI Errors

In addition to the "gross" errors we have discussed thus far, there are frequently local errors in pitch and rhythm. These errors are relative to the modifications described above. A local pitch error of $\Delta^{(P)}$ simply adds some value to the "ideal" pitch, where the ideal is determined by the relevant target note and the current transposition. A local IOI error of $\Delta^{(R)}$ has a scalar effect on the ideal IOI, derived from the relevant target note and the current tempo. Figure 1, Section c, shows examples of each error. Note that these errors do not propagate to subsequent events, and as such are termed *non-cumulative* or *local* errors. Transposition and tempo change are examples of *cumulative* error.

In some cases, there are multiple interpretations for the source of error in a query. Consider for instance Figure 2, which shows a specific interpretation of three disagreements between a target and query. The second note in the query is treated as a local pitch error of -1. The final two notes, which are a semi-tone sharper than expected (+1), are explained as a modulation. The error model, described in the next section, considers all possible interpretations, for instance considering the possibility that the error in the second note is accounted for by two modulations (before and after), and the final two notes by a pair of local errors. Depending on our expectation that such things might occur, one or the other interpretation might appear more likely. In general, we would prefer to find the most direct possible explanations for queries, since an increased likelihood of error in the model can be shown to reduce discrimination (Meek and Birmingham, 2002a).

3 Existing error models

For edits, we assume, like Mongeau and Sankoff (1990), that overall rhythm is maintained, and make the natural musical assumption that edits have a local impact on pitch. Many QBH applications adopt this approach to rhythm (Mazzoni, 2001; Meek and Birmingham, 2002b; Pauws, 2002; McNab et al., 1997, 1996).

In this study, we are concerned primarily with the distinction

between local and cumulative error. Far less is known about this area. This is largely a matter of convenience: a particular musical representation will tend to favor one approach over the other. For instance, we can adopt a pitch- and tempo-invariant representation, using pitch interval and inter-onset interval ratio (Pauws, 2002; Shifrin et al., 2002). This *relative* representation establishes a new transposition and tempo context for each note, thus introducing the implicit assumption that all errors are cumulative (Pardo and Birmingham, 2002). Pollastri (2001) determined that cumulative error is in fact far less common than local error, a conclusion supported by our studies.

Another approach to the differences in transposition and tempo context is to attempt multiple passes over a fixed context model, and evaluate error rigidly within each pass by comparing the query to various permutations of the target. Dynamic time-warping approaches (Mazzoni, 2001) and non-distributed hidden Markov model techniques (Sorsa, 2001) are well-suited to this approach. However, it is not possible to model, for instance, a modulation, using these methods, only local error. Preliminary work (Wiggins et al., 2002) uses a similar approach, grouping together "transposition vectors" connecting query and target notes. Such approaches are amenable to extensions supporting cumulative error as well, but have not – to our knowledge – been extended in this way.

Chai (2001) normalizes the tempo of the query by either automated beat-tracking, a difficult problem for short queries, or, more effectively, by giving the querier an audible beat to sing along with – a simple enough requirement for users with some musical background. Again, there is an assumption that the transposition will not change during a query, but the beat-tracker can adapt to changing tempi.

3.1 Alternative approaches

We are concerned primarily with sequence based approaches to music retrieval. Shifrin et al. (2002) relax this assumption somewhat, by translating targets into Markov models where the state is simply a characteristic relationship between consecutive notes, allowing for loops in the model. Downie (1999); Tseng (1999) model music as a collection of note n-grams, and apply standard text retrieval algorithms. In query-by-humming systems, the user is searching for a song that "sounds like..." rather than a song that is "about" some short snippet of notes, if it makes sense to discuss music in these terms at all[1]. For this reason, we believe that sequence-based methods can more accurately represent music in this context.

4 Johnny Can't Sing (JCS): A unifying model

We have developed a system supporting the simultaneous modelling of local and cumulative error known as "Johnny Can't Sing" (Meek and Birmingham, 2002b). This system provides a unique opportunity to examine the effectiveness of these two approaches, both in isolation and together. A detailed description of the training and matching algorithms used by JCS can be found in a technical report (Meek and Birmingham, 2002a).

JCS is essentially an extended hidden Markov model (Rabiner, 1989) (HMM), which associates the notes in a query with the notes in a target through a sequence of *hidden states*. The fundamental errors (transposition and tempo difference) recommend a fairly detailed state definition to describe this relationship. Each alignment of target and query notes must be considered in each of the possible tempo and transposition contexts. Consider for instance an octave-invariant representation (for instance, pitch-class): there are twelve possible transpositions, given semi-tone quantization. Further, we must model tempo differences. Consider a rhythm quantization scheme that allows for nine tempo mappings. In a song with n notes, there are thus $12 \cdot 9 \cdot n$ states, ignoring the various alignment or edit permutations.

In Figure 3.A, the conventional HMM dependency structure is shown. The hidden states (S), are each defined by a tuple, $s_i = \langle E[i], K[i], S'[i] \rangle$, and according to the first-order Markov assumption, the current state depends only on the previous state. $E[i]$ is the "Edit" type associated with the state, defining the way in which query and target notes "line up". $K[i]$ is the "Key" component, or the transposition relating the pitch in the target to the pitch in the query. $S'[i]$ is the "Speed", or the tempo mapping in the transformation.

Observations (O) are assumed to depend only on the hidden state, and are defined by $o_t = \langle Pitch, Rhythm \rangle = \langle P[t], R[t] \rangle$. Given this view of the query world, we need to determine – using machine learning techniques or by arduous hand-labelling – the probability of each combination of pitch and rhythm in the query observation given each combination of alignment, transposition and tempo in the hidden state.

It quickly becomes infeasible to explicitly model each of these states. Distributed state representations help control this complexity. The idea is to assume some degree of independence between the components of a model. The second view isolates the components of a hidden state and the components of an observation (Figure 3.B), and illustrates a more reasonable interpretation of the dependencies between these components. Only the previous edit information (E) determines the likelihood of various legal extensions to the alignment. The transposition (K) depends on both the previous transposition and the current edit type, since the degree of modulation and the current position in the target influence the probability of arriving at some transposition level. A pitch observation (P) depends only on the current edit-type and the current transposition, which tell us which pitch we expect to observe: the "emission" probability is then simply the probability of the resulting error, or discrepancy between what we expect and what we see. There is a similar relationship between the edit-type (E), tempo (S'), and rhythm observation (R).

A simple example illustrates the musical meaning of these elements. Consider the state of the model where E relates the join of the first two target notes to a query note, K is a transposition of +2 semitones, and S' is a tempo scaling of 1.25. The sequence of transformations corresponding to these components of state is shown in Figure 4, starting from the original target notes. The resulting transformed event is compared with the query event (shown in black), which is said to have a pitch error of +1 and a rhythm error, expressed as a factor, of 0.8.

5 Analysis

To maintain generality in our discussion, and draw conclusions not specific to our experimental data or approach to note representation, it is useful to analyze model entropy with respect to

[1] Beethoven's Fifth Symphony is a notable exception

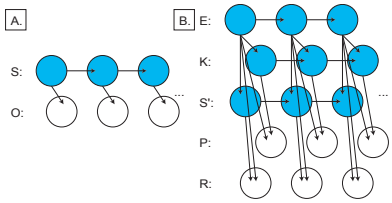

Figure 3: A possible dependency scheme for a distributed state representation.

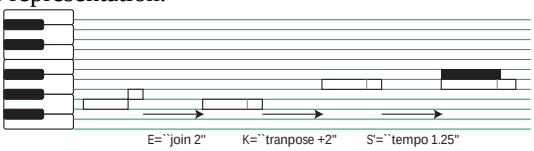

Figure 4: Interpretation of state

cumulative and local error. Intuitively, the entropy measures our uncertainty about what will happen next in the query. Formally, the entropy value of a process is the mean amount of information required to predict its outcome. When the entropy is higher, we will cast a wider net in retrieval, because our ability to anticipate how the singer will err is reduced.

What happens if we assume cumulative error with respect to pitch when local error is in fact the usual case? Consider the following simplified analysis: assume that two notes are generated with pitch error distributed according to a normal Gaussian distribution, where X is the random variable representing the error on the first note, and Y represents the second. Therefore we have: $f_X(x) = P(X = x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}$ and $f_Y(y) = P(Y = y) = \frac{1}{\sqrt{2\pi}} e^{\frac{-y^2}{2}}$. What is the distribution over the error on the *interval*? If Z is the random variable representing the interval error, we have: $Z = Y - X$. Since $f_X(x)$ is symmetrical about $x = 0$, where $*$ is the convolution operator, we have: $f_Z(z) = f_X * f_Y(z) = \frac{1}{\sqrt{4\pi}} e^{\frac{-z^2}{4}}$, which corresponds to a Gaussian distribution with variance $\sigma^2 = 2$ (as compared with a variance of $\sigma^2 = 1$ for the local error distribution). Given this analysis, the derivative entropy for local error is equal to $\frac{1}{2}(\log(2\pi\sigma^2) + 1) \approx 1.42$, and the derivative entropy of the corresponding cumulative error is roughly 1.77. The underlying distributions are shown in Figure 5. It is a natural intuition that when we account for local error using cumulative error – as is implicitly done with intervallic pitch representations – we flatten the error distribution.

While experimental results indicate that local error is most common, sweeping cumulative error under the rug can also be dangerous, particularly with longer queries. When we use local error to account for a sequence of normally distributed cumulative errors represented by the random variables $X_1, X_2, \ldots, X_n$, the local error (Z) must absorb the *sum over all previous cumulative errors*: $Z = \sum_{i=1}^{n} X_i$. For example, when a user sings four consecutive notes cumulatively sharp one semi-tone, the final note will be, in the local view, four semi-tones sharp. If cumulative error is normally distributed with variance σ^2, the expected distribution on local error after n notes is normally distributed with variance $n\sigma^2$ (a standard result for the summation of Gaussian-distributed random variables). As such, even a low probability of cumulative error can substantially effect the performance of a purely local model over longer queries.

The critical observation here is that each simplifying assumption results in the compounding of error. Unless the underlying error probability distribution corresponds to an impulse function (implying that no error is expected), the summation of random variables always results in an increase of entropy. Thus, we can view these results as fundamental to any retrieval mechanism.

6 Results

6.1 Experimental setup

160 queries were collected from five people – who will be described as subjects A-E, none involved in MIR research. Subject A is a professional instrumental musician, and subject C has some pre-college musical training, but the remaining subjects have no formal musical background. Each subject was asked to sing eight passages from well-known songs. We recorded four versions of each passage for each subject, twice with reference only to the lyrics of the passage. After these first two attempts, the subjects were allowed to listen to a MIDI playback of that passage – transposed to their vocal range – as many times as needed to familiarize themselves with the tune, and sang the queries two more times.

6.2 Training

JCS can be configured to support only certain kinds of error. For instance, it can be told to assume that only local error occurs, or only cumulative error. Regardless of the setup, JCS uses a training algorithm based on the Baum-Welch re-estimation approach (Baum and Eagon, 1970; Meek and Birmingham, 2002a). This approach learns parameters that *maximize* the expectation of the training examples, which intuitively corresponds to our goal of finding the most direct explanation possible for the errors that occur in a collection of queries. It can be shown that the procedure converges to a distribution determined by the frequency of the events being modelled, though the "events" in the hidden layer can only be interpreted indirectly. Because of the multiple hypothesis problem in the hidden layer, the optimization procedure converges to only local maxima in the search space, but by appropriately seeding the algorithm – for instance with data found by hand-labelling the training data, and with random restarts – we can find a consistent and efficient characterization of error.

The results of this training, for three versions of the model over the full set of 160 queries, are shown in Figure 6, which indicates the overall parameters for each model. For all versions, Mongeau-Sankoff-style consolidation and fragmentation are employed and result in a similar distribution: the probability of no edit is roughly 0.85, the probability of consolidation is 0.05 and the probability of fragmentation is 0.1. These values are related primarily to the behavior of the underlying note segmentation mechanism.

In one of the models, both local and cumulative error are considered, labelled "Full" in the figure. Constrained versions, with the expected assumptions, are labelled "Local" and "Cumulative" respectively. It should be apparent that the full model permits a tighter distribution over local error (rhythm error and pitch error) than the simplified local model, and a tighter distribution over cumulative error (tempo change and modulation)

than the simplified cumulative model.

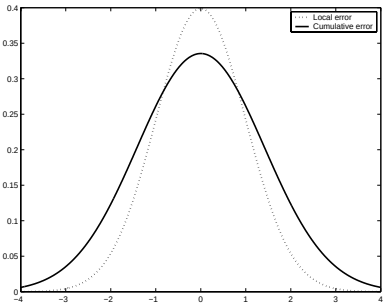

Figure 5: Assuming cumulative error when error is local

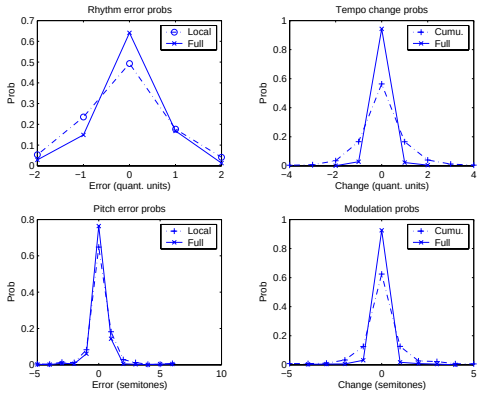

Figure 6: Result of training

When JCS has the luxury of considering both cumulative and local error, it converges to a state where cumulative error is nonetheless extremely unlikely (with probability 0.94 there is no change in tempo at each state, and with probability of 0.93 there is no modulation), which strengthens the view espoused by Pollastri (2001) that local error is indeed the critical component. This flexibility however allows us to improve our ability to predict the local errors produced by singers, as evidenced by the sharper distribution as compared with the purely local version. The practical result is that the full model is able to *explain the queries in terms of the fewest errors*, and converges to a state where the queries have the highest expectation.

6.3 Retrieval performance

Given the analysis in Section 5, it is interesting to consider the effects on retrieval performance when we assume that only local, only cumulative, or both types of error occur. To this end, we generated a collection of 10000 synthetic database songs, based on the statistical properties (pitch intervals and rhythmic relationships) of a 300 piece collection of MIDI representations of popular and classical works. In our experiments, we compare several versions of JCS:

1. 'Full' model: this version of JCS models both local and cumulative error.

2. 'Restricted' model: a version of the full model which limits the range of tempo changes and modulations ($\pm 40\%$ and ± 1 semitone respectively). This seems like a reasonable approach because training reveals that larger cumulative errors are extremely infrequent.

3. 'Local' model: only local error is modelled.

4. 'Cumulative' model: only cumulative error is modelled.

We first randomly divided our queries into two sets for training the models and testing respectively. After training each of the models on the 80 training queries, we evaluated retrieval performance on the remaining 80 testing queries. In evaluating performance, we consider the rank of the correct target's match score, where the score is determined by the probability that each database song would "generate" the query given our error model. In case of ties in the match score, we measure the *worst-case* rank: the correct song is counted below all songs with an equivalent score. In addition to the median and mean rank, we provide the mean reciprocal rank (MRR): this is a metric used by TREC (Voorhees and Harman, 1997) to measure text retrieval performance. If the ranks of the correct song for each query in a test set are $r_1, r_2, \ldots, r_n$, the MRR is equal to, as the name suggests: $\frac{1}{n}\sum_{i=1}^{n}\frac{1}{r_i}$.

The distribution of ranks is summarized in Figure 7. The rank statistics are as follows:

	Full	Restricted	Local	Cumulative
MRR	0.7778	0.7602	0.7483	0.3093
Median	1	1	1	68.5
Mean	490.6	422.9	379.5	1861

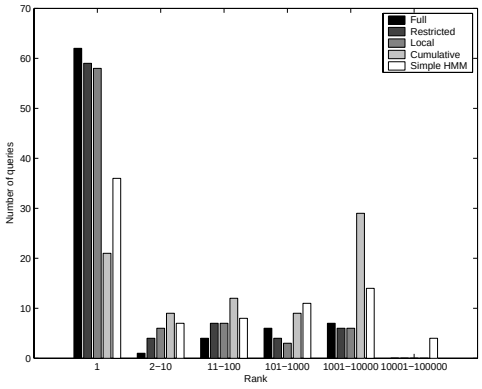

Figure 7: Distribution of ranks over real queries

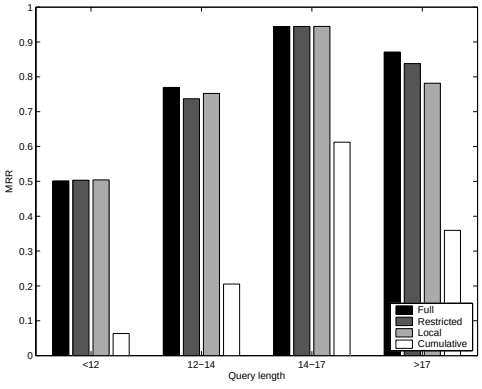

Figure 8: Retrieval performance as a function of query length

The cumulative error model performs quite poorly in comparison with the other approaches, owing to the prevalence of local error in our query collection. We see little evidence of the

reverse phenomenon: notice that restricting or ignoring cumulative error does not have a notable impact on retrieval performance except on the longest queries, where MRR decreases as we diminish the contribution of cumulative error. Figure 8 indicates this trend, where each group represents the aggregate statistics for a roughly equally-sized subset of the test queries, grouped by length. These results agree with the basic entropy analysis, which predicts greater difficulty for 'local' approaches on longer queries.

It is informative to examine where JCS fails. We identify two classes of failure:

- Alignment assumption failure: This is the most common type of error. JCS assumes that the entire query is contained in the database. When the segmenter misclassifies regions before and after the query proper as notes, this situation arises. JCS must to explain the *entire* query in the context of each target, including these margins. JCS does however model such added notes *within* the query, using the elaboration operation.

- Entropy failure: errors are so prevalent in the query that many target to query mappings appear equally strong. Interestingly, we achieve solid performance in many cases where the queries are – subjectively – pretty wildly off the mark. While using a different underlying representation might allow us to extract additional useful information from queries, this does not alter the fundamental conclusions drawn about retrieval behavior with different approaches to error.

7 Conclusions

We have demonstrated that various assumptions about the nature of errors in retrieval models can have a serious impact on performance, both in the general case through analysis, and in the specific case of the query representation used by JCS. Designers of QBH systems should consider these important interactions.

The alignment assumption failure, which will likely prove more serious in experiments with less strict controls, warrants a rethinking of our assumptions about where queries come from, and suggest a shift to local-alignment approaches, or variations thereof. In addition, it would be useful to broaden the scope of this work by examining the effects of various representations, for instance using un-quantized and un-segmented views of a query.

Acknowledgements

We gratefully acknowledge the support of the National Science Foundation under grant IIS-0085945. The opinions in this paper are solely those of the authors and do not necessarily reflect the opinions of the funding agencies. We also thank Bryan Pardo and Greg Wakefield for their comments and suggestions.

References

Baum, L. E. and Eagon, J. A. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *Annals of Mathematical Statistics*, 41:164 – 171.

Chai, W. (2001). Melody retrieval on the web. Master's thesis, Massachussetts Institute of Technology.

Downie, S. (1999). *Evaluating a simple approach to music information retrieval: conceiving melodic n-grams as text*. PhD thesis, University of Western Ontario.

Mazzoni, D. (2001). Melody matching directly from audio. In *Proceedings of International Symposium on Music Information Retrieval*.

McNab, R., Smith, L., Bainbridge, D., and Witten, I. (1997). The new zealand digital library MELody inDEX. *D-Lib Magazine*.

McNab, R. J., Smith, L. A., Witten, I. H., Henderson, C. L., and Cunningham, S. J. (1996). Towards the digital music library: Tune retrieval from acoustic input. In *Digital Libraries*, pages 11 – 18.

Meek, C. and Birmingham, W. (2002a). Johnny can't sing. Technical Report CSE-TR-471-02, University of Michigan.

Meek, C. and Birmingham, W. (2002b). Johnny can't sing: A comprehensive error model for sung music queries. In *Proceedings of International Symposium on Music Information Retrieval*, pages 124 – 132.

Mongeau, M. and Sankoff, D. (1990). Comparison of musical sequences. *Computers and the Humanities*, 24:161 – 175.

Pardo, B. and Birmingham, W. (2002). Timing information for musical query matching. In *Proceedings of International Symposium on Music Information Retrieval*.

Pauws, S. (2002). Cubyhum: a fully functional, "query by humming" system. In *Proceedings of International Symposium on Music Information Retrieval*.

Pollastri, E. (2001). An audio front end for query-by-humming systems. In *Proceedings of International Symposium on Music Information Retrieval*.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of IEEE*, volume 77 (2), pages 257 – 286.

Shifrin, J., Pardo, B., Meek, C., and Birmingham, W. (2002). Hmm-based musical query retrieval. In *Proceedings of Joint Conference on Digital Libraries*.

Sorsa, T. (2001). Melodic resolution in music retrieval. In *Proceedings of International Symposium on Music Information Retrieval*.

Terhardt, E. and Ward, W. (1982). Recognition of musical key: Exploratory study. *Journal of the Acoustical Society of America*, 72:26 – 33.

Tseng, Y. (1999). Content-based retrieval for music collections. In *ACM Special Interest Group on Information Retrieval*.

Voorhees, E. M. and Harman, D. K. (1997). Overview of the fifth text retrieval conference. In *The Fifth Text REtrieval Conference*.

Wiggins, G., Lemstrom, K., and Meredith, D. (2002). Sia(m)ese: An algorithm for transposition invariant, polyphonic content-based music retrieval. In *Proceedings of International Symposium on Music Information Retrieval*.