OPTIMAL PARAMETER ESTIMATION AND PERFORMANCE MODELLING IN MELODIC CONTOUR-BASED QBH SYSTEMS

Sumantra Dutta Roy, Preeti Rao and Rishabh Bhargava

Department of Electrical Engineering,
IIT Bombay, Powai, Mumbai - 400 076, INDIA.
{sumantra,prao,rishabh}@ee.iitb.ac.in

ABSTRACT

This paper analyses factors affecting the performance of a melodic contour-based QBH system, and presents an analytical model for the same. We present a constrained optimisation framework to find the optimal values of performance-affecting system parameters.

1. INTRODUCTION

Pitch contour is often the feature of choice for Query-by-Humming (hereafter, QBH) systems [1], [2], [3], [4], [5], [6], [7]. However, very few approaches consider evaluating the performance of QBH systems. Meek and Birmingham [8] identify different types of errors in QBH systems. Commonly used performance metrics include Precision-and-Recall [9] and Mean Reciprocal Rank (MRR) [8], [10]. Precision-and-Recall analyses involve relevancy issues, for which one either needs subjective estimates, or estimates based on a match function, which is often database- and application-specific. The Mean Reciprocal Rank (hereafter, MRR) is a measure of overall systems performance, and not just the performance to a single specific query. The rank refers to the rank of the correct item (this needs ground-truth information) in a list sorted according the a relevancy measure. Unlike the Precision-and-Recall statistics, MRR computations do not require a knowledge of database-specific information such as the total number of relevant references in an entire database.

A common representation for a musical piece is a quantised set of relative notes [4], with a convenient least count, such as a semitone. An important problem in QBH system efficiency is determining the optimum number of levels k in the relative note contour. Kageyama et al. [7] and Ghias et al. [5] use static heuristic thresholds for splitting the melodic contour into the desired number of levels. Sonoda *et al.* [6] propose the use of dynamic determination of thresholds for a 3-level contour. Kim et al. [4] consider empirical evidence to decide on choosing the number of levels of quantisation, k. To the best of our knowledge, our earlier preliminary work in this area [11], [12] are the only ones which address the above problems. Two requirements for selecting k are contradictory:

- *Fidelity:* The quantisation level determines the closeness of a match between two melodic contours. Lower the amount of quantisation, greater is the chance of two different melodic contours matching.
- *Robust Matching:* Untrained singers occasionally go off-key, so a lower quantisation level makes a system more robust in this regard.

The aim of this paper is three-fold:

1. To find the optimum quantisation level k for varying problem specifications
2. To identify parameters which affect the performance of a melodic contour-based QBH system, and
3. To propose a coefficient to evaluate the performance of such QBH systems

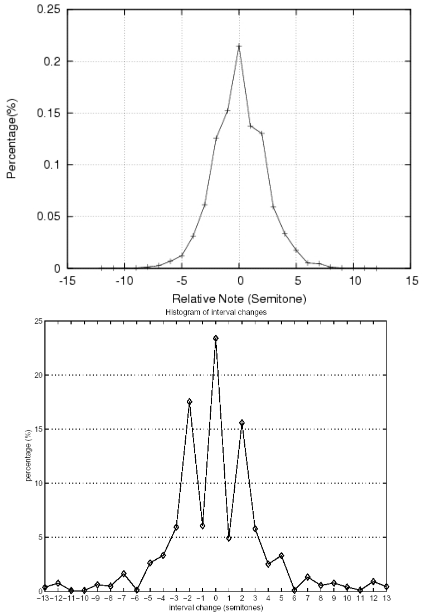

Figure 1: Distribution of relative notes: $p^D[\cdot]$ for (a) our database, and (b) an MIT database [4]

Our first work in the area [11] examines finding the optimal k for the case of uniform quantisation of the relative note axis. Our next paper [12] presents some preliminary results with extending the ideas to any general method of quantisation. We extend the ideas in our earlier papers to propose a novel coefficient to quantify the performance of a QBH system. In this paper, we experiment with some underlying basic parameters (the quantisation level k and string matching algorithms) and study their effect on the system performances: all with an actual QBH system, TANSEN [3]. We compare our results with those obtained using MRR statistics on the same QBH system.

2. THE DEMERIT COEFFICIENT

We consider a range of relative notes R. We assume that the finest level of quantisation will give rise to N relative notes, lying between two limits r_0 and r_N, respectively[1]. *Given a $N-$ level quantisation, we wish to divide this range into k intervals (using $k-1$ markers between r_0 and r_N)*

[1]The conversion between the two discrete scales $[r_0, r_N]$ and $[r'_{-N/2}, r'_{N/2}]$ is a straightforward linear and invertible function. Throughout this paper, we use the two interchangeably.

We define the **Demerit Coefficient** $\mathcal{M}_D(k,\mu)$ for a Database of songs D, as follows:

$$\mathcal{M}_D(k,\mu) \stackrel{\triangle}{=} \mu \mathcal{F}_D(k) + (1-\mu)\mathcal{R}_D(k) \qquad (1)$$

Here, $\mathcal{F}_D(k)$ and $\mathcal{R}_D(k)$ represent the **Fidelity** and **Robust-Match** functions, respectively. μ is a normalised linear combination variable. The task at hand is to find $argmin_k\ \mathcal{M}_D(k,\mu)$ i.e., that value of k and μ for which $\mathcal{M}_D(k,\mu)$ achieves a minimum value.

We assume the k intervals to be numbered 0 to $k-1$, with $k+1$ markers $m[i]$ placed according to a particular strategy: uniform quantisation (hereafter, UQ), equal probability mass (hereafter, EPMQ) [6], or heuristically placed [7], [5]. Our earlier work [12] deals with all three cases. The ith interval is characterised by the range of relative notes r_j: $m[i] \leq r_j < m[i+1]$. We define $p^D[x]$ as the discrete probability of a particular relative note x. This is a characteristic of a particular a characteristic of the particular database, and depends on its constituent songs. Fig. 1 show samples of such curves from our database of songs, as well as an MIT database [4]. We define **The Fidelity Term** $\mathcal{F}_D(k)$ as:

$$\mathcal{F}_D(k) \stackrel{\triangle}{=} \frac{1}{\widehat{\mathcal{F}_D(k)}}[\sum_{\forall\ x}[x - ind[x]]^2\ p^D[x]]^{1/2} \qquad (2)$$

Here, the summation is over all relative notes x in the songs in the given database D, and $p^D[x]$ denotes the discrete probability of a particular relative note x. We define the *Interval Indicator Function $ind[x]$* for a relative note as centroid of the interval in which it lies. $\widehat{\mathcal{F}_D(k)}$ is a normalising factor. We may take this as $\max_j |m[j+1]-m[j]|$, $0 \leq j \leq (k-1)$ for example, or simply the maximum of the terms being summed up.

We define the **Robust-Match Term** $\mathcal{R}_D(k)$ as:

$$\mathcal{R}_D(k) \stackrel{\triangle}{=} \frac{1}{\widehat{\mathcal{R}_D(k)}} \sum_{\forall\ x}[\sum_{y}(y-x)^2\ p_x^U[y]]^{1/2}\ p^D[x] \qquad (3)$$

The outer summation is over all relative notes

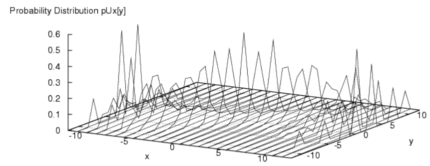

Figure 2: The probability mass function $p_x^U[y]$ for relative notes in our database

x in the songs in the given database D. The summation for y is over all relative notes which are not in the same interval as the relative note x *i.e.*, $\{r_0 \leq y < m[s]\} \cup \{m[s+1] \leq y \leq r_N\}$. $\widehat{\mathcal{R}_D(k)}$ is the normalisation factor. We may take this to be $R - \max_j |m[j+1] - m[j]|$ for instance, or simply the maximum of the terms being summed up.$p_x^U[\cdot]$ (Fig. 2) is the one-dimensional probability mass function of all user query relative notes for a given relative note x. We note that $p_x^U[\cdot]$ accounts for typical user characteristics for a given relative note x. QBH systems typically use different melodic contour matching Dynamic Programming-based approximate string matching (also referred to as the 'String Edit Distance Problem') is a widely researched area in itself [13], [14]. These typically assign a unit cost to each insertion, deletion and substitution (except in the case of an exact match, where the substitution cost is zero). The optimal match for such an algorithm gives us *a correspondence* between the notes of the query melodic string, and a reference one in the database. We use this information to build up our $p_x^U[\cdot]$ estimates.The use of the $p_x^U[\cdot]$ function also subsumes the concept of melodic similarity based on chords [15]. From Eqns. 2 and 3, we observe that the particular String Edit Distance function used will affect only the Robust Match term. In this paper, we have experimented with the Levenstein Distance (Fig. 2 uses this), the Normalised Edit Distance [14], and computational variants using heuristics specific to a QBH problem. The heuristics change the cost, and introduce computational efficiency. We use four such heuristics: to incur an appropriate cost if the query string begins before the first reference syllable, and incur no cost if: the user starts after the first reference syllable; the user ends before the last reference syllable; or continues after the last reference syllable. We examine the role of these parameters in computing the optimal value of k, in the following section.

2.1. Finding the Optimal Operating Point: The Role of Various Parameters

We need to take partial derivatives in Eqn. 1 with respect to the two variables k and μ, and set them to zero. We find the optimal k as: $k_{opt} \stackrel{\triangle}{=} \arg\min_{\forall k} |\mathcal{F}_D(k) - \mathcal{R}_D(k)|$. Further, we can find the corresponding optimal value of μ by evaluating $|\frac{\delta \mathcal{M}_D(k,\mu)}{\delta k}|$ for $\delta k = 1$ (the smallest possible discrete change in k) for $k = k_{opt}$. We can check for a minimum either using a second derivative test, or by evaluating the function itself. In case of no local minima, we check for absolute minima of the function near the boundary values of k and μ.

3. EXPERIMENTAL RESULTS

We have experimented with statistics from an existing QBH system TANSEN [3]. The reference database consists of a set of 300 songs from Hindi films. We present results of 1220 queries on the database. In Fig. 3, we find the optimal value of k for various values of the String Edit Distance. For our representative QBH system TANSEN [3], we do not observe much difference in the nature of these plots, and we get the optimal value of k to be 4 and 3 for the UQ and EPMQ cases, respectively, for the particular form of the String Edit Distance. Fig. 3 also shows the corresponding values of μ.

Fig. 4 compares our results with those of a gross measure, MRR. It is interesting to note that while the MRR peaks at $k = 9$ for the UQ case, the peak at 3 exactly corresponds to what we get through our optimisation procedure. However, we again emphasise that MRR is a gross parameter and is independent of the specific problem it is being used for, and the specific system parameters. Our analysis examines various system parameters in depth, and is the consequence of an analytical model, and optimising a suitable cost function.

4. CONCLUSIONS

This paper presents an analysis of factors affecting the performance of a melodic contour-based QBH system. We propose a performance evaluation methodology that does not involve a gross system-independent statistics such as in Precision-and-Recall and MRR. While MRR statistics also consider a match based on a distance function, our Demerit Coefficient goes much further in modelling the performance of the QBH system in terms of its parameters. We also propose a constrained optimisation framework to find the optimal values of performance-affecting system parameters.

REFERENCES

[1] C. Meek and W. Birmingham, "Johnny Can't Sing: A Comprehensive Error Model for Sung Music Queries," in *Proc. International Symposium on Music Information Retrieval (ISMIR)*, 2002.

[2] R. J. McNab, L. A. Smith, I. H. Witten, C. L. Henderson, and S. J. Cunningham, "Toward the Digital Music Library: Tune Retrieval from Acoustic Input," in *Proc. ACM Digital Libraries*, 1996.

[3] M. Anand Raju, B. Sundaram, and P. Rao, "TANSEN: A Query-By-Humming based Music Retrieval System," in *Proc. National Conference on Communications (NCC)*, 2003.

[4] Y. E. Kim, W. Chai, R. Garcia, and B. Vercoe, "Analysis of a Contour-based Representation for Melody," in *Proc. International Symposium on Music Information Retrieval (ISMIR)*, October 2000.

[5] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith, "Query By Humming - Musical Information Retrieval in an Audio Database," in *Proc. ACM Multimedia*, 1995.

[6] T. Sonoda, M. Goto, and Y. Muraoka, "A WWW-based Melody Retrieval System," in *Proc. ICMC*, October 1998, pp. 349 – 352.

[7] T. Kageyama, K. Mochiezuki, and Y. Takashima, "Melody Retrieval with Humming," in *Proc. ICMC*, 1993, pp. 349 – 351.

[8] C. Meek and W. Birmingham, "The Dangers of Parsimony in Query-by-Humming Applicaitons," in *Proc. International Symposium on Music Information Retrieval (ISMIR)*, 2003.

[9] J. L. Hsu, A. L. P. Chen, H. C. Chen, and N. H. Liu, "The Effectiveness Study of Various Music Information Retrieval Approaches," in *Proc. Conference on Information and Knowledge Management*, November 2002.

[10] R. B. Dannenberg and N. Hu, "Understanding Search Performance in Query-by-Humming Systems," in *Proc. International Symposium on Music Information Retrieval (ISMIR)*, 2004.

[11] S. Dutta Roy, P. Rao, and A. S. Galinde, "Contour-Based Melody Representation: An Analytical Study," in *Proc. National Conference on Communications (NCC)*, 2004, pp. 536 – 540.

[12] S. Dutta Roy, P. Rao, and R. Bhargava, "Melodic contour-Based QBH Sytems: Analytical Modelling and Performance Evaluation," in *Proc. National Conference on Communications (NCC)*, 2005, pp. 475 – 479.

[13] G. Navarro, "A Guided Tour to Approximate String Matching," *ACM Computing Surveys*, vol. 33, no. 1, pp. 31 – 88, 2001.

[14] G. Navarro, "Fast Computation of Normalized Edit Distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 9, pp. 899 – 902, 1995.

[15] T. C. Chou, A. L. P. Chen, and C. C. Liu, "Music Databases: Indexing Techniques and Implementation," in *International Workshop on Multi-Media Database Management Systems*, 1996, pp. 46 – 53.

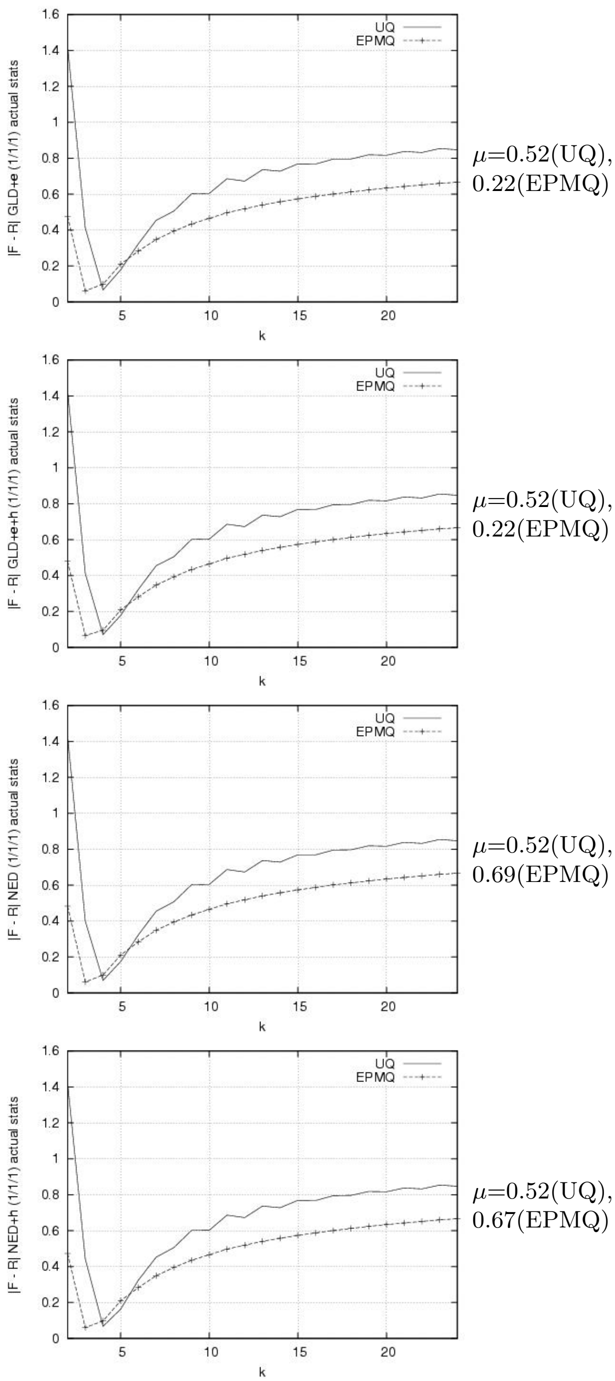

Figure 3: $|F_D(k) - R_D(k)|$ plots using UQ and EPMQ and the corresponding μ values for (from top to bottom), the Levenstein Distance: without, and with heuristics (Sec. 2); Normalised Edit Distance, without, and with heuristics, respectively. The trends in all cases are similar, $k_{opt} = 4$ (UQ) and 4 (EPMQ).

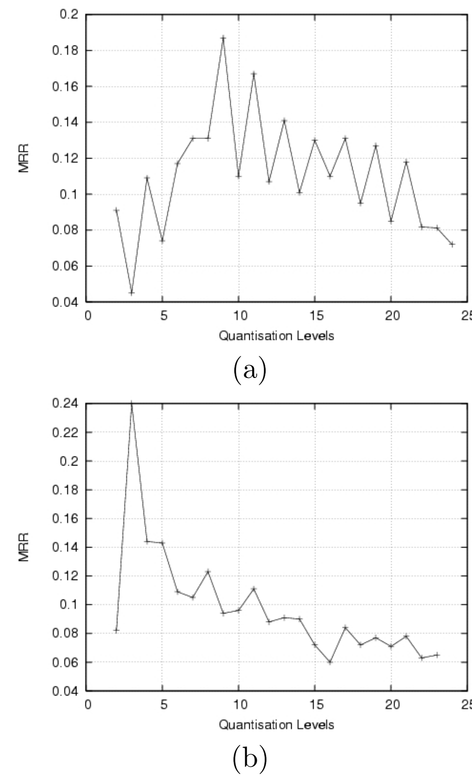

Figure 4: Variation of MRR with the number of quantisation levels for a set of 292 queries with (a) Uniform, and (b) Equal Probability Mass Quantisation, respectively.