Search-Effectiveness Measures for Symbolic Music Queries in Very Large Databases

Craig Stuart Sapp
craig@ccrma.stanford.edu

Yi-Wen Liu
jacobliu@stanford.edu

Eleanor Selfridge-Field
esfield@stanford.edu

ISMIR 2004
Universitat Pompeu Fabra
Barcelona, Spain
12 October 2004

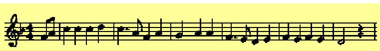
Introduction

Match-Count Profiles

Musical Features

- We examined search characteristics of 14 musical features:

7 Pitch features (*examples below*) + 7 Rhythm features: (3 duration & 4 metric)

1. duration (37~74)
2. duration gross contour (3)
3. duration refined contour (5)
4. beat level (2)
5. metric level (10~14)
6. metric gross contour (3)
7. metric refined contour (5)

pitch name	pch: F A C C C D E A F A G A A F E D E F E F E D	35 states
12-tone pitch	12p: 5 9 0 0 0 2 4 9 5 9 7 9 9 5 4 2 4 5 4 5 4 2	12
musical interval	mi: [illegible]	70/octave
12-tone interval	12i: +4 +3 0 0 +2 -2 -3 -4 +4 -2 +2 0 -4 -1 -2 +2 +1 -1 +1 -1 -2	24/octave
scale degree	sd: 1 3 5 5 5 6 5 3 1 3 2 3 3 1 7 6 7 1 7 1 7 6	7
pitch refined contour	prc: U U s s u d D D U d u s D d d u u d u d d	5
pitch gross contour	pgc: U U S S U D D D U D U S D D D U U D U D D	3

- *How do all these different features affect searching in a database?*

Anchored vs. Unanchored Searches

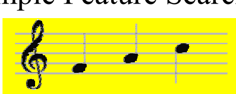
Search Pattern: F A C

Two types of search methods, Examples:

Anchored Matches search only from the start of a database entries

Unanchored Matches search starting at any position in database entries

Example Feature Searches

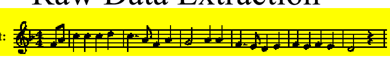


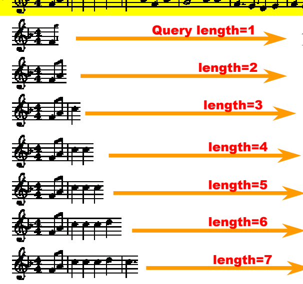
	Feature	Query *(in Themefinder)*	Anchored Matches	Unanchored Matches
pitch name	*pch*	F A C	464	1,710
12-tone pitch	*12p*	5 9 0	464	1,710
musical interval	*mi*	+M3 +m3	1,924	6,882
12-tone interval	*12i*	+4 +3	1,925	6,894
scale degree	*sd*	1 3 5	2,009	7,024
pitch refined contour	*prc*	U U	4,677	17,712
pitch gross contour	*pgc*	U U	19,787	76,865

Searching a database of 100,000 melodic incipits/themes

Raw Data Extraction

target incipit:



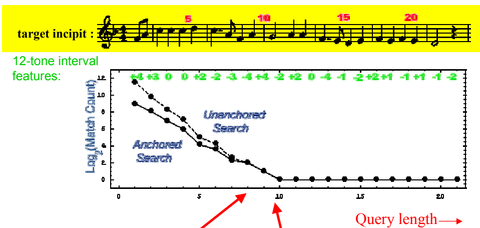
generated database queries:

10,585 matches
1,351 matches
464 matches
161 matches
83 matches
21 matches
12 matches

(anchored)

- Now plot measurements as a "*match-count profile*"

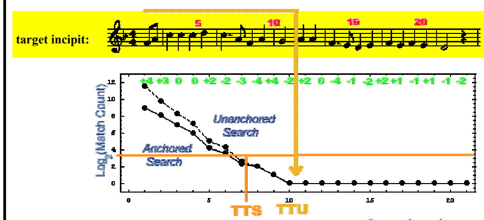
x-axis: *query length* **y-axis:** *match count (log scale)*

Individual Match-Count Profile



- Anchored and Unanchored searches merge at length = 8
- Unique match found at length = 10

Interesting Query Lengths



TTU = length of query yielding unique match

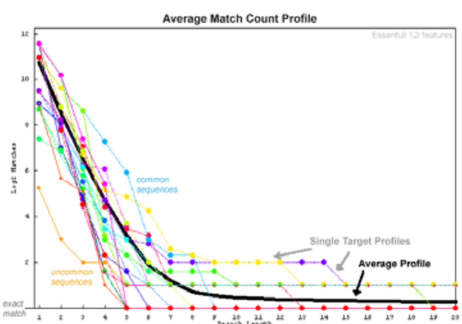
TTS = length giving matches under limit size

How long query length must be to generate a *sufficiently* small set of matches

e.g., first search-length which gives fewer than 10 matches

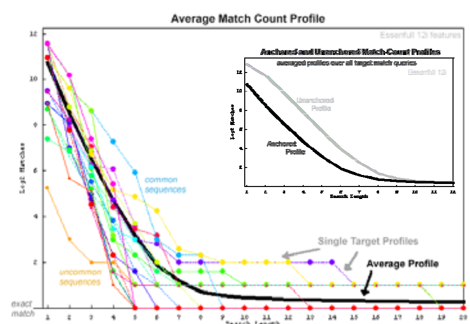
Average Match-Count Profiles

•Average all target profiles over entire database:



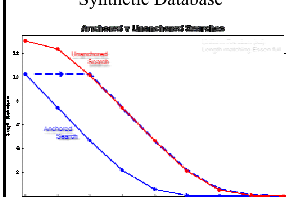
Average Match-Count Profiles

•Average all target profiles over entire database:

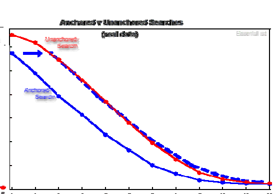


Anchored/Unanchored Profile Slopes

Synthetic Database



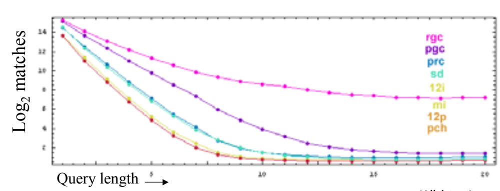
Real Database



Anchored Searching: O(log N)   Unanchored Searching: O(N^2)

- Anchored/Unanchored slopes not much different.
- Anchored searching is much faster.

Match-Count Profiles for Pitch Features

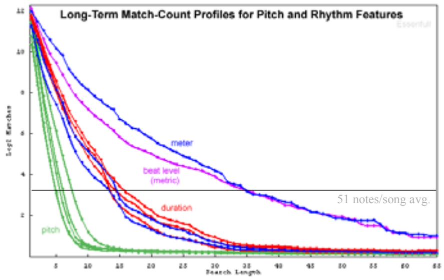


(All dataset)
18.5 notes/theme avg.

- Steeper initial slope = more descriptive feature

- **Twelve-tone pitch** and **full pitch spelling** features are very identical (orange curve)
- Absolute **twelve tone pitch** and relative **twelve-tone interval** are close.
- 7-symbol **scale degree** features close to 5-symbol **refined pitch contour**.
- 3-symbol **pitch gross contour** more descriptive than 3-symbol **duration gross contour**.

Match-Count Profiles for All Features



- TTS for rhythm twice as long than pitch TTS.
- TTS for gross metric descriptions 5 times as long as pitch TTS values.
- Rhythm feature curves more crooked.
  Phrase/meter effects?

Four Applications of Profiles:

- *Entropy & Entropy Rate*
- *Joint Feature Analysis*
- *Match-Count Predictions*
- *Synthetic Database Analysis*

Entropy

- Entropy measures basic information content of a musical feature

entropy definition:

$$H(X) \triangleq -\sum_i P_i(X) \log_2 P_i(X)$$

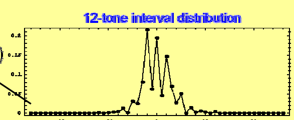
also called "Shannon Entropy" or "First-order Entropy"

Entropy (bits/symbol) — Normalized probability distribution

•Example calculation:

$$H(X) = -\sum_i P_i(X) \log_2 P_i(X)$$

$$H(12i) = 3.41165$$ bits/pitch



→ 3.4 bits/note is the minimum symbol storage size needed to store sequences of 12-tone intervals (Folksong data set).

Entropy Rate

- Entropy is a contextless (memoryless) measure.
- *Real music features are related to surrounding musical context.*
- Average entropy (entropy-rate) is more informative:

entropy-rate definition:

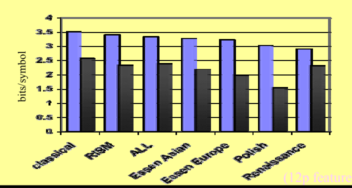
$$G(N) \triangleq \frac{H(X^N)}{N}$$

"Nth-order" entropy

also called "Average Entropy"

Entropy rate (bits/symbol) — N=Sequence length

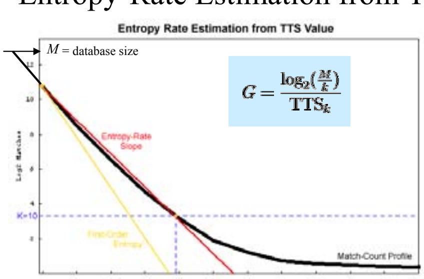
Entropy & entropy rate for various repertories:



Note:

$$G(N) \leq H(X)$$

Entropy-Rate Estimation from TTS



$$G = \frac{\log_2(\frac{M}{k})}{\mathrm{TTS}_k}$$

- *Entropy* characterizes the minimum possible average TTS.
- *Entropy-rate* characterizes the actual average TTS.

- *Entropy & Entropy Rate*
- *Joint Feature Analysis*
- *Match-Count Predictions*
- *Synthetic Database Analysis*

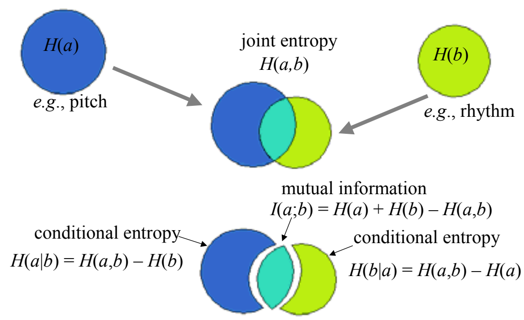
Joint Feature Analysis

Analyze
Pitch + Rhythm
as a combined feature

- How independent/dependent are pitch and rhythm features?
- What is the effect of searching pitch and rhythm features in parallel?

Mutual Information

- Measurement of the correlation of two types of features



Combining Pitch and Rhythm Searches

Individual Entropies: $H(\text{pgc}) = 1.5325$ $\quad H(\text{rgc}) = 1.4643$

Joint Entropy: $H(\text{pgc}, \text{rgc}) = 2.9900$

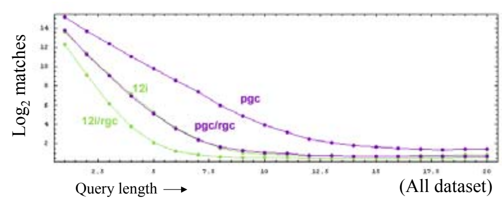
Mutual Information: $I(\text{pgc}; \text{rgc}) = H(\text{pgc}) + H(\text{rgc}) - H(\text{pgc},\text{rgc}) = 0.0068$

less than two orders of magnitude interaction

- Pitch and Rhythm are very independent features.
  (at least for pgc+rgc averaged over entire database)
- Therefore, combining independent search features should be effective.

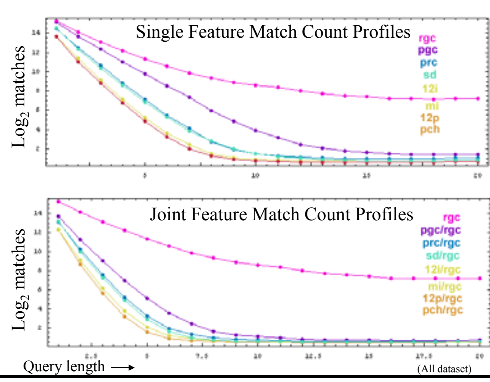
Joint Feature Profiles

for pgc/rgc vs. twelve-tone interval searching

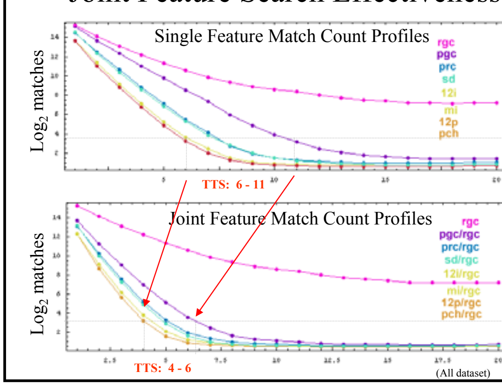


- 3*3 states work as well as 88 twelve-tone interval states.
- *pgc* and *rgc* are generic features less prone to query errors.

Joint Feature Search Effectiveness



Joint Feature Search Effectiveness



- *Entropy & Entropy Rate*
- *Joint Feature Analysis*
- ***Match-Count Predictions***
- *Synthetic Database Analysis*

Expectation Function

- Entropy Rate can be used to predict the number of matches:

$$E(n) = \frac{M}{R^n} \qquad R = 2^H$$

database size

(H = measured entropy rate)

Expected match counts for an n-length query

- Example:
  - Consider a database of "*best 3 out of 5*" Heads/Tails coin flips:

H H T H T
T H T T H
H T T H H
T T T T H
H H H H H

Entropy Rate = Entropy = $\log_2 2 = 1$ bit/symbol

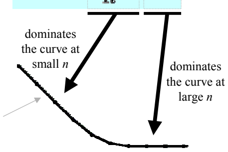
Therefore $R = 2^{\log 2} = 2^1 = 2$

- Likelyhood starting sequence is "**H**": 50% → $E(1) = M/2^1 = M/2$
- Likelyhood starting sequence is "**H T**": 25% → $E(2) = M/2^2 = M/4$
- Likelyhood starting sequence is "**H H** ": 25% → $E(2) = M/2^2 = M/4$

Match-Count Profile Constraint

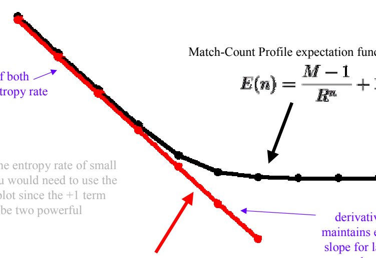
- The match-count profile queries are constructed from database entries.
- Therefore at least one match is *always expected*.
- Steal this guaranteed match from M, and add as a constant to the expectation function:

$$E(n) = \frac{M}{R^n} \rightarrow E(n) = \frac{M-1}{R^n} + 1$$



- How to get rid of curvature caused by constant +1 term?

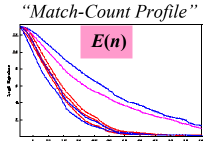
Match-Count and Derivative Profile Comparison



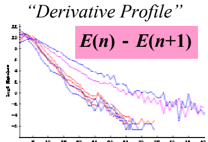
What about $E(n) - 1$?

Expectation Plot Functions

"*Match-Count Profile*"

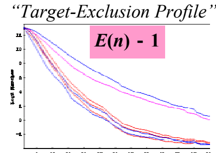


"*Derivative Profile*"



- Removes +1 curvature and *not* sensitive to duplicate entries in the database.
- Best method for measuring entropy-rate

"*Target-Exclusion Profile*"

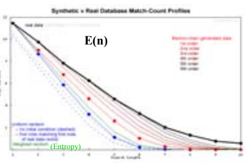


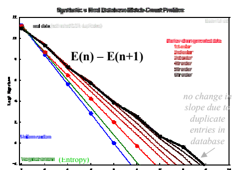
- Removes +1 curvature, but sensitive to duplicate entries in the database

Using unanchored rhythmic features for Essen-full

- *Entropy & Entropy Rate*
- *Joint Feature Analysis*
- *Match-Count Predictions*
- ***Synthetic Database Analysis***

Synthetic vs. Real Database Profiles





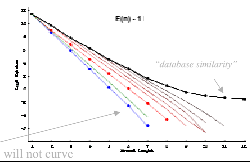
Legend:

Uniform random data

Weighted Random Based on real data probability distribution.

Markov process generated data

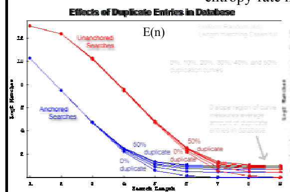
Real data

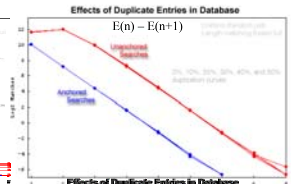


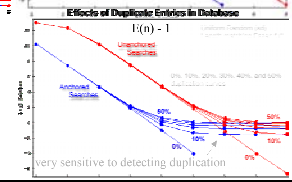
synthetic databases will not curve

Effects of Duplicate Entries on Profiles

Duplicate entries in the database do not have a significant effect on entropy-rate measurements:





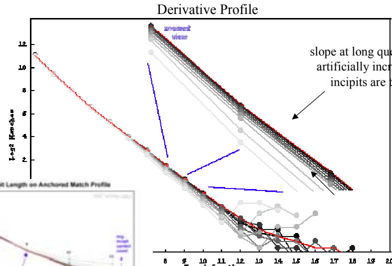


- E(n) and E(n)-1 profiles can be used to measure amount of duplication in database
- E(n) – E(n+1) removes effect of duplicate entries entirely.

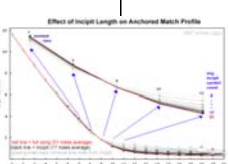
Effect of Incipit Length on Profiles

- An incipit a short initial excerpt from a full composition
- How short is is too short for a musical incipit?

Derivative Profile



slope at long query lengths is artificially increased when incipits are too short.



shorter incipits cause quantization noise in low match-count region

Search-Effectiveness Measures for Symbolic Music Queries in Very Large Databases

Craig Stuart Sapp
craig@ccrma.stanford.edu

Yi-Wen Liu
jacobliu@stanford.edu

Eleanor Selfridge-Field
esfield@stanford.edu

ISMIR 2004
Universitat Pompeu Fabra
Barcelona, Spain
12 October 2004

Summary

Interesting metrics for analyzing the effectiveness of search features:

•**Match-Count Profiles:** Examines match characteristics of a musical feature for longer and longer queries.

•**Entropy Rate:** Characterizes match count profiles well with a single number. Useful for predicting the expected average number of matches for a given length query.

•**TTS:** The number of symbols in query necessary to generate a sufficiently small number of matches (average). TTU not as useful due to noise.

Proof for Derivative Plots

$$E(n) = \frac{M-1}{R^n} + 1$$ (expectation function for Match-Count Profiles)

$$E(n) - E(n+1) = \frac{M-1}{R^n} - \frac{M-1}{R^{n+1}}$$ (subtract n and n+1 values of $E(\ )$ to cancel +1 term)

$$E(n) - E(n+1) = \frac{(R-1)(M-1)}{R\ R^n}$$ (algebra manipulation)

plotting on a log scale, so take the log of both sides:

$$\log_2[E(n) - E(n+1)] = \log_2\left[\frac{(R-1)(M-1)}{R}\right] - \log_2 R^n$$

Let: $y = \log_2[E(n) - E(n+1)]$ and $b = \log_2\left[\frac{(R-1)(M-1)}{R}\right]$

so the equation becomes: $y = b - \log_2 R^n$

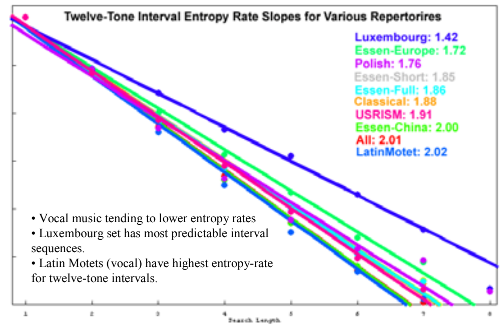
$y = b - \log_2 2^{Hn}$ since $R = 2^H$

Let: $x = n$

$$\underbrace{y = -Hx + b}_{\text{slope}}$$

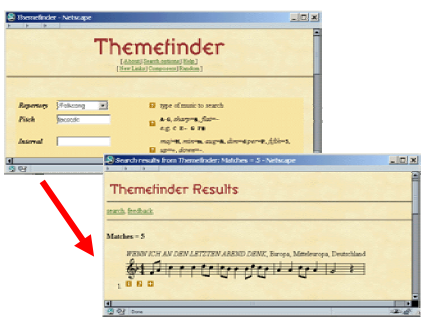
which is a line with a slope proportional to the entropy (rate)

Derivative Plots for 12i features



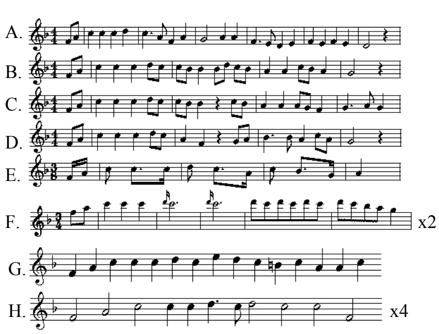
Themefinder Website

http://www.themefinder.org

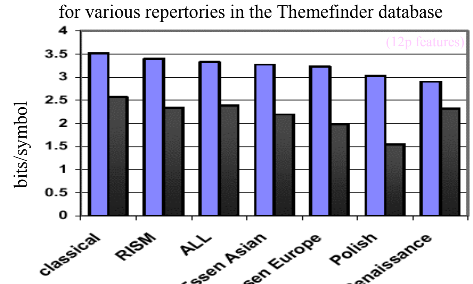


Themefinder Collections

Data set	Count	Web Interface
Classical	10,718	themefinder.org
Folksong	8,473	themefinder.org
Renaissance	18,946	latinmotet.themefinder.org
US RISM A/II	55,490	
Polish	6,060	
Luxembourg	612	lux.themefinder.org
total:	100,299	

Matches on First Seven Notes

Entropy and Entropy Rate

for various repertories in the Themefinder database

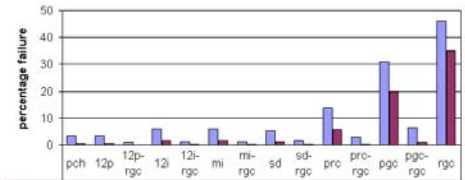


$G(N) \leq H(X)$ Entropy rate less than or equal to the Entropy

Search Failure Rates

Database size: 100,299

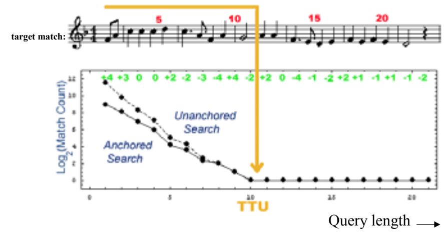
Average note count/incipit: 16



	pch	12p	12p-rgc	12i	12i-rgc	mi	mi-rgc	sd	sd-rgc	prc	prc-rgc	pgc	pgc-rgc	rgc
TTU	3.4	3.4	1.1	5.9	1.4	5.9	1.4	5.3	1.5	14	2.8	31	6.5	46
TTS	0.53	0.53	0.11	1.5	0.15	1.5	0.16	1.3	0.21	5.8	0.38	20	1.1	35

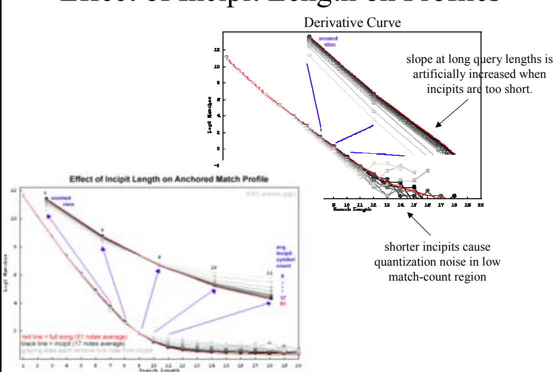
•Plot measures how often a search produces too many matches for query sequences as long as the database entry.

Time To Uniqueness

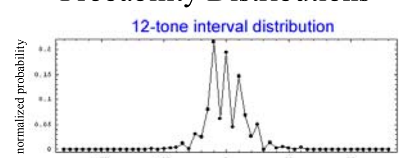


TTU = the number of query symbols needed to find the exact match in the database. Turns out to not be very useful since it is more susceptible to noise in the data.

Effect of Incipit Length on Profiles



Probability Distributions



$$H(X) = -\sum_i P_i(X) \log_2 P_i(X)$$

$$H(12i) = 3.41165$$

3.4 bits/note is the lower symbol storage size limit needed to store sequences of 12-tone intervals (Folksong data set).

• Entropy can be used as a basic estimate for how many notes are necessary to find a unique/sufficient match in the database, but ...

Expectation Function

M = database size

$E(n)$ = average expected match counts for an *n*-length query

$R = 2^H$ where H is the entropy rate of the feature being searched for (Entropy rate is assumed to be constant)

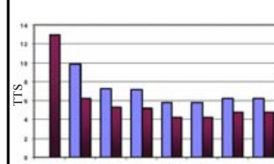
In general: $E(n) = \dfrac{M}{R^n}$

For example, consider sequences created with a uniform random distribution of three states (the next symbol in the sequence is equally likely to be any of the three states). Then, the entropy of the sequence is: $H = \log_2 3$ which makes $R = 2^{\log_2 3} = 3$

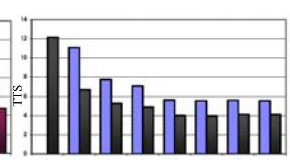
and the formula for the expected match counts becomes: $E(n) = \dfrac{M}{3^n}$

then 1/3 of the database entries should be matched with a one-length query on the average: $E(1) = \dfrac{M}{3^1} = \dfrac{M}{3}$ and a length-two query should return 1/9 of the database on the average: $E(2) = \dfrac{M}{3^2} = \dfrac{M}{9}$

Joint Pitch/Rhythm Effects on TTS



Chinese Folksongs dataset



Classical dataset

•Adding *rgc* to pitch features usually reduces the search length by 2 notes.

•Combining *rgc* and *pgc* reduces search length by 4 notes.