Multiresolution Rhythm Beat Induction Model

From Emcap

Contents

- 1 MultiresRhythm
  - 1.1 What Is The Module?
  - 1.2 What Are The Inputs/Outputs?
  - 1.3 What Are The Parameters?
  - 1.4 What Are The System Requirements?
  - 1.5 How To Cite
  - 1.6 How To Use
    - 1.6.1 Salience Trace Input File Format
    - 1.6.2 Onset Input File Format
    - 1.6.3 Output File Format
  - 1.7 Example Of Use

MultiresRhythm

by Leigh M. Smith <lsmith@science.uva.nl>

Music Cognition Group, ILLC/Universiteit van Amsterdam

2008/1/28

MultiresRhythm code

What Is The Module?

This is the Common Lisp implementation of MultiresRhythm, a simulation of the cognition of musical rhythm using a representation derived from continuous wavelet transforms for multiresolution time-frequency analysis and interpretation. See (Smith 1999, Smith & Honing 2008a, Coath et. al 2007). A portion of this model is used to represent rhythmic expectation (Smith & Honing 2008b) and is implemented as a standalone command-line executable module that can be interfaced to other EmCAP modules.

What Are The Inputs/Outputs?

Inputs consist of two different formats. The first form consists of times and likelihoods of onsets of notes in seconds. The second form consists of a continuous salience trace sampled at a rate of 200Hz. Output consists of a list of expectations of future events represented as times of expected events, a confidence measure of the expectation of each time and an estimated precision of the estimated time of next onset.

What Are The Parameters?

A file containing a description of the rhythm derived from onset processing of the audio signal. The command line specifies filenames, sample rate, and format of data, either a continuous saliency trace, or as onset times.

What Are The System Requirements?

1. GSL - Gnu Scientific Library (http://www.gnu.org/software/gsl/).
2. SBCL - A Common Lisp implementation that has been tested and is recommended (http://www.sbcl.org).
3. ASDF - Another System Definition Facility (included in SBCL).
4. CFFI - C foreign function library for Common Lisp (http://common-lisp.net/project/cffi/).
5. NLISP - Numerical modelling and plotting library (http://www.nlisp.info).
6. CL-CLI-PARSER - Command line option parsing library (http://common-lisp.net/project/cl-cli-parser)

- Install GSL
  - On Linux install libgsl using Debian apt or RedHat rpm.
  - On MacOS X install gsl with fink:

```
% fink install gsl
```

- Install SBCL
  - On Linux install sbcl using Debian apt or RedHat rpm.

- On MacOS X install sbcl with fink:

```
% fink install sbcl
```

- Install CL-CLI-PARSER.
  - CL-CLI-PARSER is currently not an ASDF-Install'able package.
  - Download the latest version to a directory using CVS.
  - Create symbolic links to it in .sbcl/systems:

```
% ln -s ~/your_dir/cl-cli-parser/cli-parser.asd ~/.sbcl/systems/cli-parser.asd
```

- Install MultiresRhythm package
  - Unarchive the rhythm package:

```
% unzip MultiresRhythm-2.7.zip
```

-
  - Create symbolic links to it in .sbcl/systems:

```
% ln -s ~/your_dir/MultiresRhythm_2.7/multiresrhythm.asd ~/.sbcl/systems/multiresrhythm.asd
```

If you don't want to (or can't) create the symlinks, you can tell ASDF within Lisp where to find them prior to running (require 'multiresrhythm):

```
CL-USER> (pushnew "/path/to/your/multiresrhythm.asd" asdf:*central-registry*)
```

- Install libraries.

Run SBCL, load ASDF and ASDF-Install, install CFFI and then NLISP.

```
% sbcl
CL-USER> (require 'asdf)
CL-USER> (require 'asdf-install)
CL-USER> (asdf-install:install 'cffi)
CL-USER> (asdf-install:install 'nlisp)
CL-USER> (require 'multiresrhythm)
```

- Generate a stand alone executable.

Run SBCL from the command line (not within SLIME) and generate the executable with:

```
CL-USER> (multires-rhythm:generate-executable)
```

This will create a large single executable binary named "emem-expect-mrr"

How To Cite

@InProceedings{my:icmpc08,

```
 author =        {Leigh M. Smith and Henkjan Honing},
 title =         {A Multiresolution Model of Rhythmic Expectancy},
 booktitle = {Proceedings of the Tenth International Conference on Music Perception and Cognition},
 year =          2008,
 address =       {Sapporo, Japan},
 note =          {(in press)}
```

}

@Article{coath:nips07,

```
 author =        {Martin Coath and Susan Denham and Leigh M. Smith and Henkjan Honing and
                 Amaury Hazan and Piotr Holonowicz and Hendrik Purwins},
 title =         {An auditory model for the detection of perceptual onsets and beat tracking in singing},
 booktitle =    {Connection Science},
 year =          2008,
 note =          {(submitted)}
```

}

@InProceedings{my:icomcs07,

```
 author =        {Leigh M. Smith and Henkjan Honing},
 title =         {Evaluation of Multiresolution Representations of Musical Rhythm},
 booktitle =    {Proceedings of the International Conference on Music Communication Science},
 editor =       {Emery Schubert and Kym Buckley and Rosemary Eliott and Brooke Koboroff
```

```
                 and Johnson Chen and Catherine Stevens},
annote = {ISBN 978-1-74108-161-9},
address =       {Sydney, Australia},
note =          {Published online as \url{http://marcs.uws.edu.au/links/ICoMusic/Full_Paper_PDF/Smith_Honing.pdf}},
year =           2007
```

}

@Article{smith:jmm07,

```
author =        {Leigh M. Smith and Henkjan Honing},
title =         {Time-Frequency Representation of Musical Rhythm by Continuous Wavelets},
journal =       {Journal of Mathematics and Music)},
volume =       2,
number =       2,
note =          {(in press)},
year =          2008
```

}

How To Use

The expectation generation is run as a command-line application with the input and output represented as files. The input file format is selected with a command line option. The usage is:

```
Usage: emem-expect-mrr [options]
```

Where *[options]* are:

- -t, --onset-times[=ARG] Whether the input format consists of onset times. e.g --onset-times=music.onsets
- -s, --saliency-trace[=ARG] The input file in a format consisting of sampled saliency traces. e.g --saliency-trace=music.saliency
- -r, --sample-rate=ARG The sample rate (in Hz) used for sampled saliency traces (see option saliency). e.g --sample-rate=[200]
- -o, --output-file[=ARG] The name of the file for expectancy times. e.g --output-file=expectancy

Salience Trace Input File Format

The file format consists of samples in arbitrary units, one per line in ASCII floating point format. The sample rate is assumed to be 200Hz (suitable and typical for rhythm representation).

For example:

```
0.234
0.783
1.002
1.178
0.888
```

Onset Input File Format

The file format currently described is proposed for ease of reading, rather than flexibility. In practical terms, both the input and output formats are intended to be highly similar and describe structured parameterised note lists similar to the MusicKit (Jaffe & Boynton 1989, Brandon & Smith 2000) or MusicXML. These offer the advantage that they allow additional parameters to be incorporated without major rewrites of the I/O of each connecting module. At the moment, we refrain from specifying representations in those formats to reduce implementation time and to establish whether it is necessary to move to a binary file format to reduce I/O time. Note that the semantic meaning of each feature is not specified, it is implicitly dependent on the order of the features and is a major limitation of the current non-XML format.

@Article{jaffe:next,

```
author = {David A. Jaffe and Lee R. Boynton},
journal = cmj,
number = 2,
pages = {48--55},
title = {An Overview of the Sound and Music Kits for the {NeXT} Computer},
volume = 13,
year = 1989
```

}

@InProceedings{brandon:newworld,

```
author =        {Stephen Brandon and Leigh M. Smith},
title =         "Next Steps from {NeXTSTEP}: {MusicKit} and {SoundKit} in a
                New World",
booktitle =  {Proceedings of the 2000 International Computer Music
                Conference},
pages =         {503--6},
year =  2000,
address =       {Berlin},
organization = {International Computer Music Association},
annote =        "http://www.musickit.org/Publications/ICMC2000.pdf"
```

}

The input file format consists of note onsets, one per line in ASCII floating point format, space separated. Each line specifies the tuple:

```
ONSET-TIME PHENOMENAL-ACCENT ONSET-TIME-VARIANCE;FEATURE1-VALUE,FEATURE1-VARIANCE FEATURE2-VALUE,FEATURE2-VARIANCE ... FEATUREn-VALUE,FEAT
```

Where:

- ONSET-TIME = Time of onset in seconds.
- PHENOMENAL-ACCENT = A measure of onset event accent, such as created by intensity or timbre. Specified in arbitrary units.
- ONSET-TIME-VARIANCE = The span of time that the onset is considered to occur within (one standard deviation either side of ONSET-TIME).
- FEATUREn-VALUE = Feature of the onset such as amplitude or timbral intensity. Specified in arbitrary relative units.
- FEATUREn-VARIANCE = The variation in the feature.

For example:

```
0.000000,0.000000;0.429396,0.000000 0.371519,0.000000
0.371519,0.000000;0.511888,0.000000 0.327982,0.000000
0.699501,0.000000;0.586299,0.000000 1.039093,0.000000
1.738594,0.000000;0.553394,0.000000 0.165442,0.000000
```

Output File Format

Expectancies produced by emem-mrr-expect are specified in the file with respect to a given time point, by enclosing in XML-like "EXPECT" delimiters, identified by the time point. Each line within the EXPECT tag specifies one expected time for a future event, and it's likelihood:

```
<EXPECTANCIES>
<EXPECT ID="event-index" TIME="time-in-seconds">
EXPECTED-TIME CONFIDENCE PRECISION;FEATURE1-VALUE,FEATURE1-VARIANCE FEATURE2-VALUE,FEATURE2-VARIANCE FEATUREn-VALUE,FEATUREn-VARIANCE
</EXPECT>
</EXPECTANCIES>
```

Where:

- event-index is an integer distinguishing each moment that expectations are generated from.
- time-in-seconds is a floating point value in seconds.
- EXPECTED-TIME = The estimated time of onset in seconds.
- PRECISION = A measure of the degree of accuracy of time estimation in seconds, effectively the variance around the estimated time.
- CONFIDENCE = A relative measure of likelihood that the event is expected at the time.
- FEATUREn-VALUE = A measure of the expected value of the given feature.
- FEATUREn-VARIANCE = A measure of the expected variance of the given feature around the value.

For example:

```
<EXPECTANCIES>
<EXPECT ID="0" TIME="29.36500">
29.48315 0.16667 0.01115;1.0,0.23 0.49,0.345
29.76239 0.12500 0.04129;0.9,0.14 0.94,0.32
30.00500 0.12500 0.09630;0.8,0.34 0.43,0.30
33.86101 0.08333 0.03845;0.95,0.3 0.4222,0.13
36.60577 0.07639 0.07846;0.45,0.2 0.41,0.95
31.61300 0.18750 0.21449;0.32,0.02 0.76,0.13
</EXPECT>
</EXPECTANCIES>
```

Example Of Use

The command line tool can be checked for the version:

```
% emem-expect-mrr
```

The expectancy file is generated from a file of onsets with:

```
% emem-expect-mrr -t file.onsets -o file.expectancy
```

The expectancy file is generated from a file of salience trace and explicitly setting the sample rate with:

```
% emem-expect-mrr -t file.onsets -s file.expectancy -r 200.0
```

Retrieved from "http://iua-share.upf.es/wikis/emcap/index.php/Multiresolution_Rhythm_Beat_Induction_Model"

- This page was last modified 11:23, 12 August 2008.