

# pYIN — pitch and note tracking in monophonic audio - Bug #1997

## Possible bug in probability normalization

2020-07-01 10:51 AM - Benjamin van Niekerk

<b>Status:</b>	New	<b>Start date:</b>	2020-07-01
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			

### Description

Hi,

I noticed a possible bug at <https://code.soundsoftware.ac.uk/projects/pyin/repository/entry/YinUtil.cpp#L310> when normalizing the peakProb probabilities. When  $i = \text{minInd}$ ,  $\text{peakProb}[\text{minInd}]$  gets normalized so for every  $i > \text{minInd}$  we multiply by a smaller value. As a result the final sum won't add to  $\text{peakProb}[\text{minInd}]$  and the probability that a frame is voiced would be lower than expected. However, this bug is also useful: if you run pYIN on a pure tone, for example, all the minima will have the same probability so the wrong pitch might be selected. This bug will reduce the probabilities for all minima after  $\text{minInd}$ .

### History

#### #1 - 2020-07-02 09:54 AM - Chris Cannam

Interesting discovery, thanks for reporting it!

This certainly doesn't **look** intentional, but perhaps Matthias can weigh in. I'd love to know what difference it makes in terms of performance - I wonder if we can set up a test. Definitely not something we can just fix without testing, especially if, as you say, it might have benefits.

#### #2 - 2020-08-05 09:45 PM - Matthias Mauch

Chris Cannam wrote:

*Interesting discovery, thanks for reporting it!*

*This certainly doesn't **look** intentional, but perhaps Matthias can weigh in. I'd love to know what difference it makes in terms of performance - I wonder if we can set up a test. Definitely not something we can just fix without testing, especially if, as you say, it might have benefits.*

Hi Benjamin, Chris,

I've looked into this a little, and I have a few observations

- it does indeed seem to be a bug
- the bug stems from a commit that was made in a hurry (by me) and it also includes some other big, ugly code changes that resulted in the implementation diverging from the pYIN algorithm as devised and tested in our paper

I think the right thing would be to re-instate the code that is currently commented out and actually reflects pYIN from the paper. I will see if I am allowed to do this given my current employment in industry. In the meantime, feel free to contact me via my private email.

@Benjamin: if you, by any chance, have code and ground truth lying round that would quickly allow you to assess the accuracy of any changes, that would be really handy. — I don't at the moment, but if I'm allowed to work on this I might try.

Cheers,  
Matthias.

### #3 - 2020-08-06 11:51 AM - Benjamin van Niekerk

Thanks for the reponse Chris and Matthias.

As a bit of context, I discovered the bug when doing some comparisons against a version of pYIN I implemented for librosa. I'm guessing the goal of the changes implemented by Matthias was to allow pitch candidates other than the first minimum of the difference function? Based on this idea, the approach I went with in librosa was to put a prior over the position of the minimum. I used a discrete Boltzmann distribution that puts most of the probability mass on the first minimum and then less on successive minima. By changing the temperature of the distribution you can move between the two approaches: at one extreme you recover the implemetation described in the paper and at the other, you get a uniform distribution over all minima. In tests I ran on the MDB-stem-synth dataset this seemed to work quite well (see <https://github.com/librosa/librosa/pull/1063#issuecomment-651042308> for example).

I'd be happy to do some benchmarking on MDB-stem-synth if that'd be helpful.

### #4 - 2020-08-16 11:56 AM - Matthias Mauch

Benjamin van Niekerk wrote:

*Thanks for the reponse Chris and Matthias.*

*As a bit of context, I discovered the bug when doing some comparisons against a version of pYIN I implemented for librosa. I'm guessing the goal of the changes implemented by Matthias was to allow pitch candidates other than the first minimum of the difference function? Based on this idea, the approach I went with in librosa was to put a prior over the position of the minimum. I used a discrete Boltzmann distribution that puts most of the probability mass on the first minimum and then less on successive minima. By changing the temperature of the distribution you can move between the two approaches: at one extreme you recover the implemetation described in the paper and at the other, you get a uniform distribution over all minima. In tests I ran on the MDB-stem-synth dataset this seemed to work quite well (see <https://github.com/librosa/librosa/pull/1063#issuecomment-651042308> for example).*

*I'd be happy to do some benchmarking on MDB-stem-synth if that'd be helpful.*

Hi Benjamin,

I'm very impressed with the contribution you made, and the thread you pointed to is amazing. I've also thanked Brian separately. The Boltzmann distribution was in fact the thing that made me stumble most when looking at your implementation, so I'm glad I understand it now in principle.

Now that I understand your approach I'm even tempted to port it back to the Vamp implementation. But as a first step it'd be nice to simply tidy up there. I'll try to do that and also see if I can implement at least a clean switch between the two "extremes". Then, if I may, I would love to come back to your offer to do the benchmarking. (I do still really like the method as described in the paper — one of the more elegant things I've done — and it would be great to see how it does.)

Thank you for your work and explanation!  
Matthias.

### #5 - 2020-08-20 09:42 AM - Benjamin van Niekerk

Hi Matthais,

Glad to have helped! Yeah, the pYIN paper is great and for the most part I've been setting the Boltzmann distribution temperature towards the direction described in the paper.

*But as a first step it'd be nice to simply tidy up there. I'll try to do that and also see if I can implement at least a clean switch between the two "extremes".*

That sounds good to me. Happy to do the benchmarking, just let me know when you're ready.