The Viterbi algorithm used on continuous gravitational waves

Written by Athene Demuth (GLF136), Andreas Hermansen (PHC250) and Rune Kutchinsky (LXP458).
Niels Bohr Institute - Write Up for Advanced Methods in Applied statistics
(Dated: 8. March, 2023)

Review of the article "SOAP: A generalised application of the Viterbi algorithm to searches for continuous gravitational waves", written by Joe Bayley, Chris Messenger, and Graham Woan.

PRELIMINARY DISCUSSION

This discussion pertains to time series. These are any series of data that vary in time, ie. plotted as $[f(t), t]$. The signals discussed here are periodic, with a slowly varying frequency. To investigate this, Short (time) Fourier Transforms (**SFT**) are utilized. Conjoined, short, time intervals are individually Fourier transformed, to give a spectral history.
More specifically, the data looked at in the article are artificial measurements of slowly varying continuous gravitational waves made by LIGO. The injected signal amplitude is of the same order as the background noise, necessitating sophisticated signal analysis. Template comparison is often used for this. Here, very specific signal shapes predicted by theory are tested and varied till one fits satisfactorily. This gives optimal precision, but at very large computational cost. The SOAP group implement an algorithm based on a hidden Markov model: the Viterbi algorithm, to significantly reduce the computational cost, as described later in this proceeding. Several important approximative steps are taken, for while the physical system being measured is (presumed) continuous, any digital measurement is necessarily discrete. Amplifiers have rise times, and only finite amplitude resolution. This data is then lumped into even bigger time chunks as a by the SFT, which also has maximal and minimal frequency, and -resolution. The *ansatz* is then that the physical system is well approximated by this set of discrete time and frequency states, which is also what the Viterbi algorithm takes as input. This is valid for sufficiently tight frequency and time resolutions.

PRELIMINARY THEORY

In the article, all the data is grouped into equal length time-intervals Δt, for which the signal is reasonably approximated to have a constant frequency, and Gaussian noise is added to the signal. Performing Fast Fourier Transform (FFT) on each time slice and squaring gives a spectrogram for the Power Spectral Density (PSD) - also known as SFT.

Hidden Markov models

This treatment of the data is made to allow for a hidden Markov model (HMM) interpretation of the data. In hidden Markov models, a Markov model can not be directly observed, but only observed through emission probabilities. The hidden Markov model dictates the frequency for each group of data, and then a slice ν_j of the signal is observed. By assuming the data follows a HMM the probabilities can be written using Bayes's Theorem and the Markov property as:

$$\max_{\vec{\nu}} P(\vec{\nu}|D) = \frac{P(D|\nu)P(\nu)}{P(D)} \tag{1.1}$$

$$P(\vec{\nu}) = P(\nu_0) \prod_{j=1}^{N-1} P(\nu_j|\nu_{j-1}) \tag{1.2}$$

$$\ln P(\vec{\nu}) = \ln P(\nu_0) + \sum_{j=1}^{N-1} \ln P(\nu_j|\nu_{j-1}) \tag{1.3}$$

Where D is the data, and $\vec{\nu} = \{\nu_0, \nu_1, \dots \nu_n\}$ will be referred to as a track, ν_j denotes the location of the Markov model at time j. The probabilities $P(D|\nu)$ and $P(\nu_n|\nu_{n-1})$ are found by assuming a special structure of the HMM. For convenience $P(D|\nu)$ is known as the emission probabilities $P(\nu_n|\nu_{n-1})$ are known as the transition probabilities. $P(\nu_0)$ denotes the probability distribution over the initial state. Now the problem becomes a discrete maximization problem, where the maximization happens over the set of all possible tracks. An efficient algorithm known as the Viterbi algorithm computes the most likely track given the data. The Viterbi algorithm requires knowledge of the assumed structure of the HMM.

THE VITERBI ALGORITHM

As previously described, the Viterbi algorithm finds the most probable track trough the time-frequency grid. In its most simple form, the Viterbi algorithm uses data from a single detector, and the next step depends only on the previous position, as per the Markov property. To keep it simple, the track is limited to staying in the same frequency bin (Center - C), going up (U), or one bin down (D) in the next time step. This can be implemented by using a transition matrix, which stores the prior probability for each direction. In the case of transitions limited to UCD, the transition matrix ($\ln P(\nu_j|\nu_{j-1})$ rescaled) will contain three numbers, and could be written as a stencil: $[0, 1, 0]$, to favor the center transition.

Given the prior (transition matrix) and the log-likelihood grid, the Viterbi algorithm is readily applied. Figure 1 shows the starting point for the algorithm. Notice that we have gone from probabilities to log likelihood (**LLH**), and since we omit the marginal likelihood $P(D)$, the value is scaled.

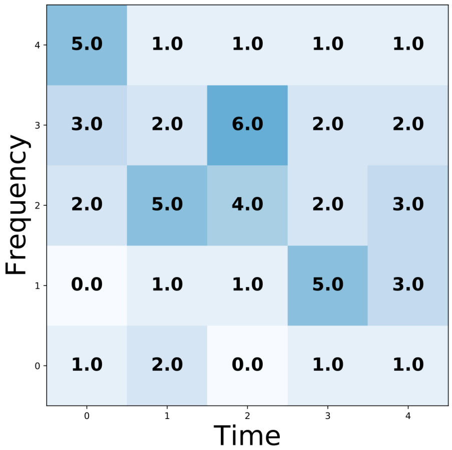

Figure 1: Scaled LLH for each time segment and frequency bin.

Figure 2 shows the track recorded from the Viterbi algorithm. For the first segment of time, there is no prior, and the previous transition is set to center. For each of the next segments of time, the maximum likelihood is found for each frequency, given the possible previous positions. The cumulative sum over the previous position's scaled LLH plus any bonuses from the transition matrix is then taken, and added to the value at the given frequency. The best possible previous position is then stored, as shown in figure 2. This is done sequentially for all time slices. Finally, the last time slice is used to find the most probable track, starting with the highest cumulative scaled LLH frequency bin, the route can be followed backwards using the stored transitions.

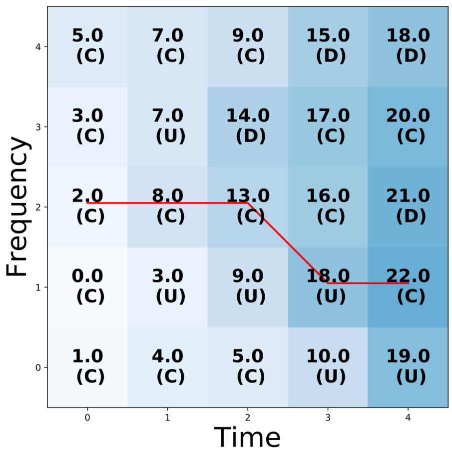

Figure 2: Track through the scaled LLH grid.

SNR OPTIMIZATIONS

Outlining a few developments of the base algorithm discussed in the article: Rather than just looking at the previous step, older steps can also be taken into account. This however leads to a significant increase in computational cost. A more advanced transition matrix (or stencil) can also be implemented, so as to incorporate more advanced physics into the path selection. This is much more problem specific, and also increases the computational complexity of the algorithm, though not as much as the 'memory' implementation.

TESTING THE VITERBI ALGORITHM

To test their Viterbi algorithm, the SOAP team used it on multiple different simulated and real data sets. Figure 3 shows a simulated data sets for two detectors. And even though no line seems to be present, the Viterbi algorithm finds a track, shown in the bottom part of the figure.

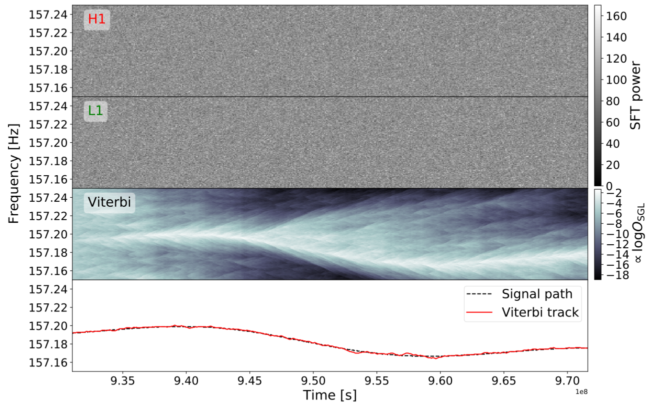

Figure 3: The two upper pictures are frequencies measured from two detectors. The third shows a variant of the LLH ratio, which is normalized along the y-axis. The lower picture shows the most probable track found by the Viterbi algorithm, along with the injected signal.

Results

The difference between the track found with the Viterbi algorithm in figure 3 and the input signal gives an RMS of about 1 frequency bin. The Viterbi algorithm computation scales linearly in the size of the spectrogram, which is an improvement over the previous method of template fitting. Generally, they obtain a RMS of ~ 2 between the Viterbi track and the injected signal.

OTHER APPLICATIONS OF VITERBI ALGORITHM

Historically the Viterbi algorithm was designed to decode a noisy bit-channel. It has been applied to a broad range of problems, from Natural Language Processing to DNA sequencing.

LINKS — REFERENCES

(All last visited 03/08/23)
http://www.scholarpedia.org/article/Viterbi_algorithm
https://ieeexplore.ieee.org/document/1054010/
https://journals.aps.org/prd/abstract/10.1103/PhysRevD.100.023006