Signal Processing Algorithms & Architecture
Dr. Anirban Dasgupta
Department of Electronics & Electrical Engineering
Indian Institute of Technology Guwahati

Lec 4: Peak Detection

Hello all, welcome to this new lecture on peaks. Detection and smoothing as part of the course. Signal processing algorithms and architectures. This is Dr. Anirban Dasgupta and let us begin. So, in time-domain signal processing, we have Several problems, and we have covered things.

Like time shifting and scaling, amplitude shifting. and scaling, modulation, correlation, Convolution and zero-crossing detection. So, today we are going to focus on the peak. Detection, and we will also cover topics.

Like envelope detection, smoothing, and Pattern matching will be covered in subsequent lectures. So, what is peak detection? So, this is a technique in signal processing. To identify local maxima and minima. A signal, and these are common things. Like, what is a peak and what is a trough? So, a peak is basically a point in the signal.

which is higher than its immediate neighbours. And a trough is a point in the signal that is. lower than its immediate neighbours. So why is this important? So, these peaks and troughs can represent. some significant events in the signal.

I will give you some real-life examples. So, what is the problem with peak detection? So, this problem is to identify the peaks. Troughs and the specific locations in time. So, as per the definition, a point $x[n]$ It should be considered a peak if it is greater than its previous sample $x[n-1]$ and $x[n]$ plus that is the following sample. Similarly, a point will be considered a trough if it is less than $x(n - 1)$, that is the previous sample or the following sample $x[n+1]$. Mathematically,

A point $x[n]$ is considered a peak if
$x[n] > x[n-1]$ and $x[n] > x[n+1]$
A point $x[n]$ is considered a trough if
$x[n] < x[n-1]$ and $x[n] < x[n+1]$

So, this is a pictorial representation of the peaks. and troughs in a signal. The peaks are represented by red dots and troughs They are represented by green dots.

So, what are the applications? So, if you look at audio signal processing, this has a good

application in beat detection. So, whenever there is a beat, there will be. Peaks in the audio signal, and this is useful. In tempo analysis, rhythm recognition, and Synchronization with visual effects in media. Also, it is useful for detecting pitch in audio signals or typically sound waves. And this is useful in music composition. software, auto tuning and sound analysis. In biomedical signal processing, peak detection has A hell of a lot of applications in ECG signals. For example, there is something called RPICS.
which is happening in the QRS complex, and this is the highest point in the QRS complex. And this peak detection or detection of the R peaks help one to analyze the heartbeats and important parameters. Like heart rate variability, so you can diagnose diseases such as arrhythmia, or it is also used in wearable devices.

Like we have our smartwatches and this. Heart rate calculation is using some sort of peak detection algorithm. EEG signals, for example, are another area where peak detection is useful.

Specifically, to find brainwave patterns like alpha waves, beta waves, and theta waves. And this is helpful in the analysis of sleep studies. Like, how is the quality of your sleep? Do you have epilepsy or not, or a cognitive state Monitoring, such as how cognitively active you are, whether you are facing some kind of fatigue or tiredness, so these are used.

Peak detection at the back end in vibration monitoring and fault detection, this is useful, like analyzing mechanics. Vibrations or predicting faults in machines. So, this is also a very big application. A lot of faults are happening in machines.

The industries waste a lot of money in this process in maintaining those machines. So, if the fault can be detected in advance using, some vibration signal or maybe another signal using some sensors, so these devices or these machines can be protected or given proper care before any fault occurs.

In seismology, earthquake detection and peak detection is a vital parameter specifically for detection of earthquake, to locate and quantify the earthquakes. Also in volcanic activity monitoring, to monitor the volcanic activity such as an eruption or a magma movement. So, these are just the tips of the iceberg.

There are a lot of applications of peak detection. So, let us now try to understand what. are the peak detection methods. So, the first method is thresholding, which is kind of a naive method, then thresholding using the first derivative, using the second derivative and the windowing method, like Sliding window to find local maxima or minima. Now, in this terminology, it often peaks and troughs are collectively called peaks.

So let us try to understand the algorithms. So, starting with thresholding, in this method, thresholds are set, and any value of a signal if it exceeds the threshold, Say I have some signal peaks like this. So, to me, this is probably not a good peak. So if I set a threshold like this, anything above the threshold, if I have this condition

$$x[n] > x_{threshold}$$

and also, the immediate neighbor check like this point is greater than its neighbor Points, then I will say this point is a peak. So, there can be a lot of spurious peaks in the lower end of the signal. So, these peaks will not be detected if we are using a simple thresholding method. So, there are n comparisons because you. First, compare with the threshold, and you compare with the previous one and the upcoming samples. So the complexity will be $O(n)$. So here you see a lot of peaks are detected using this thresholding method and the threshold actually decides how many peaks will be detected. To me, this looks like a wave where there are two prominent peaks. This is one region of peak. This is one region of peak. But because of the noise, there are a lot of spurious peaks. So, we can change the threshold, and by increasing it, we see that with a low threshold, many peaks are detected.

As I increase the threshold, a lot of spurious peaks are removed, and as I increase it further, I get a smaller number of peaks, and with a threshold of 1, I get very few peaks. Maybe a threshold of 1.5 will give me exactly 2 peaks that I am looking for. So, what are the limitations of this method? So, although this method is very easy to understand, this method is highly sensitive to noise. So, if there are small fluctuations, they will be identified as peaks, which may not even signify any physical phenomena in the original signal.

Then the fixed threshold problem is just like in zero crossing detection with a fixed window size. So here, the fixed threshold problem may be difficult to detect peaks, especially for non-stationary signals. So here, probably some adaptive thresholding will be beneficial. Then, missed peaks, like if the threshold is too high, may be missed. Or, if the threshold is too low, you can have too many peaks.

So this is very much dependent on the threshold, and this is good enough to understand, but for applying it in practical scenarios, you may need to do something else. One of them is smoothing because the peak is a high-frequency component. Of course, I have not introduced the concept of frequency yet. We are in the time domain. So I will try my best to restrict myself in the time frame.

But consider that there are peaks like these. So, maybe this is only one big peak, but there are small oscillations across the peak that will satisfy the threshold condition as well as the neighborhood condition. So, if I smooth this signal like this. You end up with only one peak, which is the desired peak. So, again, as I said, the smoothing techniques we will discuss later, but from the english word, you understand that smoothing is something that removes this abrupt fluctuation.

So, the general idea is to apply a smoothing filter, like a moving average filter, a Gaussian filter, or any low-pass filter you can apply, and then use the threshold-based detection method. So, this is the algorithm. So, you first smooth the signal using one of the filters discussed, and then you apply the threshold. Now, let's analyze the complexity. So, we have discussed that only $O(n)$ complexity is required for thresholding.

But when we are smoothing, typically we are convolving with a filter of size w. And this is what I am considering: we are using a linear filter, which means we are smoothing the signal by the convolution operation. So, in such a case, the convolution will take a complexity of $O(N \cdot W)$ i,e

$$O(N \cdot W) + O(N) = O(N \cdot W)$$

where W is the filter size, and again, this is the naive way of convolving or the brute force method of convolving. If I use fast fourier methods, then the complexity can become $log(w)$, $\log_2 w$ instead. Now what we are doing is thresholding with smoothing.

Now you can see that after thresholding and smoothing, we see a much smaller number of peaks or false peaks, I would say, compared to the signal without smoothing. And in one case, like in this one, because of the thresholding operation, We are even reduced to only two peaks, but these two. They are not the actual or desired peak; maybe here we desire. Only one peak, but we are getting two peaks. Of course, this can be mitigated by even further smoothing.

But that is not the point; this peak is missed. So, selecting the threshold is again creating the problem, and this is where the question arises about blurring peaks. So, your actual peak may be very subtle and because of this smoothing process your peak may be blurred, and because of the choice of threshold, your peak may be missed. So this threshold problem can be adjusted by using an adaptive threshold that will dynamically adjust itself based on the signal characteristics. Such as the signal mean or standard deviation, this will allow you to detect peaks with greater precision because the threshold will now be variable.

Based on the local neighborhood of the signal, the algorithm of adaptive thresholding is that first we will smooth the signal and then we will use adaptive. Thresholding based on the properties of the signal, and I would say local properties, means based on some small window. And if you compare these three methods, such as just using a fixed threshold, using threshold smoothing and threshold. Using an adaptive threshold, of course, with smoothing. So, you will find that at each stage, there is an improvement.

So, these are the things. Now again, in this case, we have taken several parameters of thresholding. Now, to me, this is the best place where I am actually locating all the true peaks. Of course, there are two false peaks; here we get two peaks where there should be one. But for the other cases, like these two cases, the peaks are entirely missed, and in this case, we get only one peak. So, we can do some post-processing, like comparing the distance between two peaks, and then we can boil down to the actual required peaks.

So the advantages are that it is robust to noise because of the smoothing factor, as well as because of this changing amplitude. By using dynamic thresholding and local statistics, we are able to detect peaks better. or peaks in a better manner. But the limitations and challenges is the window size selection based on the local statistics? Which I was talking about, so what window size it should be and how much. To take that is one question here, and that is typically dependent on your exact signal or the application.

Then tuning the parameters, such as whether the standard deviation is a good measure. Parameter, or shall we use some other parameter for selecting the threshold? A big question, and of course, this does not address. The issues with edge, specifically if you are using windows. So, at windows, there will be a lot of edges. So, for each window, there will be two boundary values or edge values, so how to handle them is again a question.

So one way is to mirror the signal by using a small portion of the boundary region and mirror them and see whether you get a peek or not. Now let us come to the derivative method, which is analogous. To the analog method in calculus, where we use our $f(x)$ derivatives to find the maxima and minima.

So the concept is that. When we have a peak or a trough, what happens is that my derivative? So if this is a peak, then there is an increasing part and a decreasing part.

So, what happens is, if you see the slope, the slope before. This and slope after this are opposite polarities. So that is if the slope at the previous point and at that point. Specific point, or you can also take one point ahead at $n + 1$. They will be of different polarities, and you can use the product to find.

Whether they are of different signs, and then you can mark them. As a peak or not, mark that specific point as a peak and the second derivative method is just like the local maximum method not local maxima, but the maxima method used in calculus, where you do. The second derivative test checks whether the second derivative is negative, which indicates that it is a maxima. Why will it be negative? Because my signal is increasing and so that first derivative will be positive and then after the peak, my signal will be decreasing.

Which means the first derivative will be negative. So, which means that my rate is changing from being positive to negative. So, hence my double derivative will be negative and similarly, you can also detect a trough because this is the same algorithm, just with the polarities and signs reversed, so I am not explaining it explicitly or you can consider the peaks and troughs in signal processing concepts are synonymously used as peaks, so you can take the absolute value of the signal and then apply all these techniques and at last we are interested to detect local maxima. So, this is again, you can take a small window, and in that window w you can slide it and use the same technique. So, this technique is basically dividing the whole signal into sub-signals and applying the methods.

On each sub-signal and then finally merge the peaks. Now this window may be overlapping and may not be overlapping. So overlapping windows may solve some boundary problem issues, but complexity wise it may not be very good; so if you are doing a real-time application, It is not a very good way to take a lot of windows and also to take highly overlapped windows. So if you are taking non-overlapping windows and you have the provision for parallel processing, then that is still very good. But with overlapped windows, you can also do parallel processing; however, you will still get a lot of windows.

Specifically, if you are going for a very high overlap, like 90 percent overlap or 75 percent overlap. So, thank you so much. So these are the concepts of peak detection. We will get back to you with other interesting topics. Thank you so much.