VITERBI ALGORITHM

Article *in* Journal of Experimental Algorithmics · October 2015

CITATIONS
0

READS
1,921

1 author:

Davidrichesemmanuel Akkidas

5 PUBLICATIONS **0** CITATIONS

SEE PROFILE

VITERBI ALGORITHM

LOVELY PROFESSIONAL UNIVERSITY
MR.DAVID RICHES
Department of Electronics and Communication Engineering

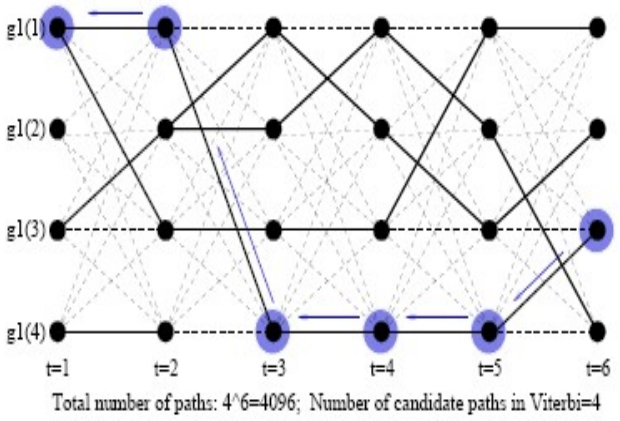
ABSTRACT

The Viterbi algorithm (VA) is a recursive optimal solution to the problem of estimating the state sequence of a discrete-time finite-state Markov process observed in memoryless noise. Many problems in areas such as digital communications can be cast in this form. This paper gives a tutorial exposition of the algorithm and of how it is implemented and analyzed. Applications to date are reviewed. Increasing use of the algorithm in a widening variety of areas is foreseen. This paper is a tutorial introduction to the Viterbi Algorithm, this is reinforced by an example use of the Viterbi Algorithm in the area of error correction in communications channels. Some extensions to the basic algorithm are also discussed briefly. Some of the many application areas where the Viterbi Algorithm has been used are considered, including it's use in communications, target tracking and pattern recognition problems. A proposal for further research into the use of the Viterbi Algorithm in Signature Verification is then presented, and is the area of present research at the moment

KEYWORDS-Algorithm design and analysis, Convolutional codes, Decoding, Digital communication, Helium, Markov processes, Recursive estimation, State estimation, Stochastic processes, Viterbi algorithm.

INTRODUCTION

The Viterbi Algorithm (VA) was first proposed as a solution to the decoding of convolutional codes by Andrew J. Viterbi in 1967, with the idea being further developed by the same author in.It was quickly shown by Omura that the VA could be interpreted as a dynamic programming algorithm. Both Omura and Forney showed that the VA is a maximum likelihood decoder. The VA is often looked upon as minimizing the error probability by comparing the likelihoods of a set of possible state transitions that can occur, and deciding Which of these has the highest probability of occurrence. A similar algorithm, known as the Stack Sequential Decoding Algorithm (SSDA), was described by Forney in as an alternative to the VA, requiring less hardware to implement than the VA. The SSDA has been proposed as an alternative to the VA in such applications as target tracking and high rate convolutional decoding. It can be shown though, that this algorithm is sub-optimum to the VA in that it discards some of the paths that are kept by the VA.

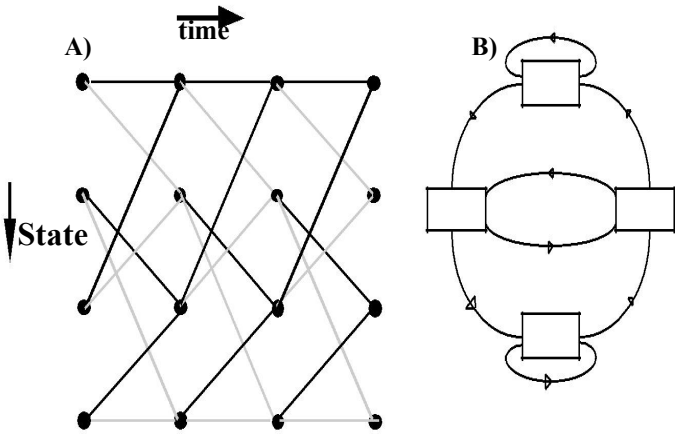


Since it's conception the VA has found a wide area of applications, where it has been found to be an optimum method usually out performing previous methods. The uses it has been applied to not just covers communications for which it was originally developed, but includes diverse areas such as handwritten word recognition, through to nonlinear dynamic system state estimation.

This report is in effect a review of the VA. It describes the VA and how it works, with an appropriate example of decoding corrupted convolutional codes. Extensions to the basic algorithm are also described. In section 3 some of the applications that the VA can be put to are described, including some uses in communications, recognition problems and target tracking. The area of dynamic signature verification is identified as an area requiring further research.

THE ALGORITHM

The VA can be simply described as an algorithm which finds the most likely path through a trellis, i.e. shortest path, given a set of observations. The trellis in this case represents a graph of a finite set of states from a Finite States Machine (FSM). Each node in this graph represents a state and each edge a possible transitions between two states at consecutive discrete time intervals. An example of a trellis is shown below in Figure 1a and the FSM that produced this trellis is shown in Figure 1b. The FSM re-ferred to here is commonly used in digital electronics and is often referred to in the lit-erature as a Markov Model (MM)



Showing a) trellis diagram spread over time and b) the corresponding state diagram of the FSM

For each of the possible transitions within a given FSM there is a corresponding out-put symbol produced by the FSM. This data symbol does not have to be a binary digit it could instead represent a letter of the alphabet. The outputs of the FSM are viewed by the VA as a set of observation symbols with some of the original data symbols corrupted by some form of noise. This noise is usually inherent to the observation channel that the data symbols from the FSM have been transmitted along.

The trellis that the VA uses corresponds to the FSM exactly, i.e. the structure of the FSM is available, as is the case in it's use for convolutional code decoding. Another type of FSM is the Hidden Markov Model (HMM). As the name suggests the actual FSM is hidden from the VA and has to be viewed through the observations produced by the HMM. In this case the trellis's states and transitions are estimates of the under-lying HMM. This type of model is useful in such applications as target tracking and character recognition, where only estimates of the true state of the system can be pro-duced. In either type of model, MM or HMM, the VA uses a set of metrics associated with the observation symbols and the transitions within the FSM. These metrics are used to cost the various paths through the trellis, and are used by the VA to decide which path is the most likely path to have been followed, given the set of observation symbols.

Before defining the VA the following set of symbols have to be defined:-

t - The discrete time index.

N - Total number of states in the FSM.

xn - The nth state of the FSM.

ot - The observation symbol at time t, which can be one of M different symbols.

sp_{nt} - The survivor path which terminates at time t, in the nth state of the FSM. It consists of an ordered list of xn's visited by this path from time t = 0 to time t.

T- Truncation length of the VA, i.e. the time when a decision has to be made by the VA as to which sp_{nt} is the most likely.

πn - Initial state metric for the nth state at t = 0. Defined as the probability that the nth state is the most likely starting start, i.e. Prob(x_n at t = 0).

a_{nm} - The transition metric for the transition from state x_m at time t - 1 to the state xn at time t. Defined as the probability that given that state xm occurs at time t - 1, the state xn will occur at time t, i.e. Prob(x_n at t | x_m at t - 1).

bn - The observation metric at time t, for state xn. Defined as the probability that the observation symbol ot would occur at time t, given that we are in the state xn at time t, i.e. Prob(o_t | x_n at t).

Γnt - The survivor path metric of sp_{nt}. This is defined as the Product of the metrics (πn,a_{nm} and b_n) for each transition in the nth survivor path, from time t = 0 to time t.

The equations for the model metrics, π_n ,a_{nm} and b_n, can be derived mathematically where their

properties result from a known application. If the metric properties are not known, re-estimation algorithms can be used, such as the Baum-Welch re-estimation algorithm, to obtain optimum probabilities for the model. It is also usual to take the natural logarithm of the metrics, so that arithmetic underflow is prevented in the VA during calculations.

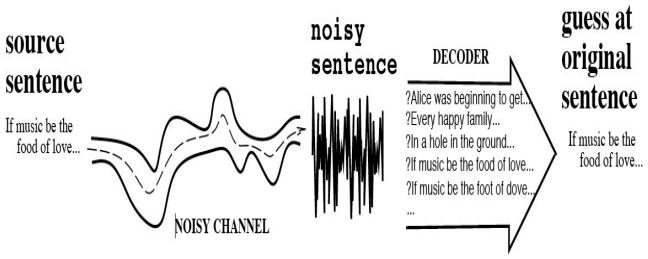
The VA can now be defined:-

Initialization. t = 0;
For all n, where 1 to N
$\Gamma n0 = \ln \pi n$;
$sp_{n0} = [xn]$;
End For;
Calculation:
For all t, where 1 t T,
For all n, where 1 n N
For all m, where 1 m N
$\Gamma_{nt} = \text{Max} [\Gamma_{mt}\text{-}1 + \ln a_{nm} + \ln b_n]$;
End For;
sp_{nt} = Append[xn,spmt] such that $\Gamma_{mt}\text{-}1 + \ln a_{nm} + \ln b_n = \Gamma_{nt}$;
End For;
End For;
Decision.
If t = T,
For all n, where 1 n N $\Gamma_T = \text{Max} [\Gamma_{nt}]$;
End For;
$sp_T = sp_{nt}$ such that $\Gamma_{nt} = \Gamma_T$;

In English the VA looks at each state at time t, and for all the transitions that lead into that state, it decides which of them was the most likely to occur, i.e. the transition with the greatest metric. If two or more transitions are found to be maximum, i.e. their met-rics are the same, then one of the transitions is chosen randomly as the most likely transition. This greatest metric is then assigned to the state's survivor path metric, $\Gamma_{nt.}$ The VA then discards the other transitions into that state, and appends this state to the survivor path of the state at t - 1, from where the transition originated. This then becomes the survivor path of the state being examined at time t. The same operation is carried out on all the states at time t, at which point the VA moves onto the states at t + 1 and carries out the same operations on the states there. When we reach time t = T (the trun-cation length), the VA determines the survivor paths as before and it also has to make a decision on which of these survivor paths is the most likely one. This is carried out by determining the survivor with the greatest metric, again if more than one survivor is the greatest, then the most likely path followed is chosen randomly. The VA then outputs this survivor path, sp_T, along with it's survivor metric, Γ_T.

EXAMPLE

Now that the VA has been defined, the way in which it works can be looked at using an example communications application. The example chosen is that of the VA's use in convolutional code decoding, from a memoryless Binary Symmetric Channel (BSC), as described. This consists of encoding the input sequence, transmitting the sequence over a transmission line (with possible noise) and optimal decoding the sequence by the use of the VA.



The input sequence, we shall call it I, is a sequence of binary digits which have to be transmitted along the communications channel. The convolutional encoder consists of a shift register, which shifts in a number of the bits from I at a time, and then produces a set of output bits based on logical operations carried out on parts of I in the register memory. This process is often referred to as convolutional encoding. The encoder introduces redundancy into the output code, producing more output bits than input bits shifted into it's memory. As a bit is shifted along

the register it becomes part of other output symbols sent. Thus the present output bit that is observed by the VA has information about previous bits in I, so that if one of these symbols becomes corrupted then the VA can still decode the original bits in I by using information from the previous and subsequent observation symbols.. It is assumed here that the shift register only shifts in one bit at a time and outputs two bits, though other combinations of input to output bits are possible.

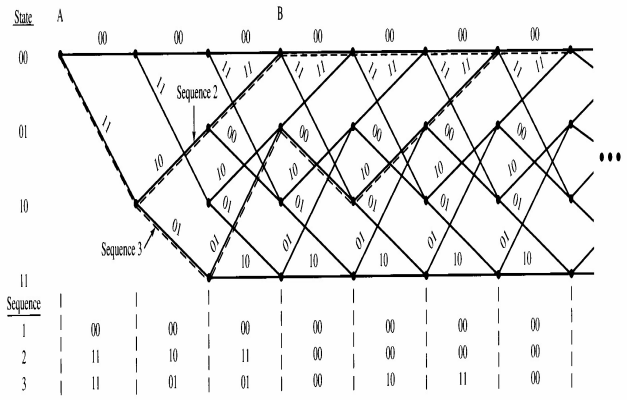
This encoder can be represented by the FSM .This state corresponds to the actual contents of the shift register locations S2 followed by S1, i.e. if we are in state 01, then the digit in S1 is 1 and the digit in S2 is 0. The lines with arrows, represent the possible transitions between the states. These transitions are labeled as x/y, where x is a two digit binary number, which represents the output symbol sent to the communications channel for that particular transition and y represents the binary digit from I, that when shifted into the encoder causes that particular transition to occur in the state machine.

The encoded sequence produced at the output of the encoder is transmitted along the channel where noise inherent to the channel can corrupt some of the bits so that what was transmitted as a 0 could be interpreted by the receiver as a 1, and vice versa. These observed noisy symbols are then used along with a trellis diagram of the known FSM to reconstruct the original data sequence sent.. This shows the states as the nodes which are fixed as time progresses. The possible transitions are shown as grey lines, if they were caused by a 1 entering the encoder, and the black lines, if they were caused by a 0 entering the encoder. The corresponding outputs that should of been produced by the encoder are shown, by the two bit binary digits next to the transition that caused them.

It was shown by Viterbi in that the log likelihood function used to determine survivor metrics can be reduced to a minimum distance measure, known as the Hamming Distance. The Hamming distance can be defined as the number of bits that are different between, between the symbol that the VA observes, and the symbol that the convolutional encoder should have produced if it followed a particular input sequence. This measure defines the combined measure of a_{nm} and b_n for each transition in the trellis. The πn's are usually set before decoding begins such that the normal start state of the encoder has a $\pi n = 0$ and the other states in the trellis have a πn whose value is as large as possible, preferably π. In this example the start state of the encoder is always assummed to be state 00, so $\pi 0 = 0$, and the other πn's are set to 100.

ALGORITHM EXTENSION

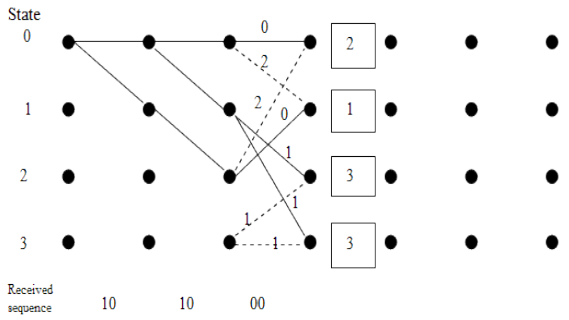
Now that the VA has been examined in some detail, various aspects of the algorithm are now looked at so that a viable research area can be established. In this section, some of the possible extensions to the algorithm are looked as though these are limited in scope.



In the example above the VA relies on inputs from a demodulator which makes hard decisions on the symbols it received from the channel. That is, whatever type of modulation used, be it phase, frequency or amplitude modulation, then the demodulator has to make a firm decision whether a 0 or 1 was transmitted. One obvious extension of the VA is that of replacing this firm decision with a soft-decision. In this method the demodulator produces soft outputs, i.e. each symbol produced by the demodulator in-stead of consisting of a 0 or 1 consists of the symbol that the demodulator thinks was sent along with other bits which represent the confidence that the symbol was transmit-ted. This increases the information presented to the VA increasing it's performance. The VA can then be used to decode

these soft decisions and output a hard decision as before.

The next step up from this is a Viterbi algorithm which produces soft output decisions of it's own- this is known as a Soft Output Viterbi Algorithm (SOVA). In this version of the VA a hard decision is given on the most likely survivor path, but information about how confident that each symbol in the path occurred is also produced.



Another extension to the algorithm was suggested recently, by Bouloutas. Bouloutas's extension generalizes the VA so that it can correct insertions and deletions in the set of observations it receives, as well as symbol changes. This method combines the known FSM that produced the symbols in the first place, such as in the example above, with an FSM of the observation sequence. A trellis diagram is produced, known as a product trellis diagram, which compensates for insertions and deletions in the observation sequence. For each of the insertion, deletion and change operations a metric is assigned, which also depends upon the application the VA is being applied to. The VA produces the most likely path through the states of the FSM, estimates of the original data sequence, as well as the best sequence of operations performed on the data to obtain the incorrect observation sequence. An application of this VA, is in the use of correcting programming code whilst compiling programs, since many of the errors produced while writing code tend to be characters missed out, or inserted characters. Another extension to the VA is that of a parallel version. Possible solutions to this have been suggested by Fettweis and Meyr , and Lin *et al* . The parallel versions of the VA have risen out of the needs for fast hardware implementations of the VA.

CONCLUSION

The Viterbi Algorithm has been described and various extensions to this have been covered. It was noted that the Viterbi Algorithm could be used on unknown finite state machines, referred to as Hidden Markov Models. Also, it was found that the Viterbi Algorithm has a wide field of applications, and not just in communications for which it was first developed. Particular attention was paid to its application to target tracking and dynamic non-linear equations, and it has been suggested that this could be applied to the on-line verification of handwritten signatures. A thesis proposal was also given, concentrating on the application of the VA and HMM's to dynamic signature verification. Suggestions were also given, where appropriate, for further research in other areas which can be viewed as either extensions to the thesis proposal or beyond the scope of the proposed research.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. They are also grateful to Professor Chris Clarke, Darren Kerbyson for theoretical analysis.

REFERENCES

[1] Viterbi, A.J., Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm, IEEE Transactions on Information Theory, April 1967; IT - 13 (2) : pp. 260 - 269.

[2] Viterbi, A.J., Convolutional Codes and Their Performance in Communication Systems, IEEE Transactions on Communications Technology, October 1971; COM - 19 (5): pp. 751 – 772

[3] Omura, J.K., On the Viterbi Decoding Algorithm, IEEE Transactions on Information Theory, January 1969; IT - 15 (1): pp. 177 -179.

.