Andrew J. Viterbi

A Personal History of the Viterbi Algorithm

I have never considered myself a digital signal processing (DSP) practitioner, but over nearly half a century of a career in wireless communication involving both spatial and terrestrial systems, I have often employed DSP as a principal tool of my trade. After all, signals are the essence of communication and, thanks to the evolution of Moore's law, digital processing of communication signals has progressed from a curiosity in the 1950s to the only sensible implementation of the communication receiver's baseband functions today.

EDITORS' INTRODUCTION

Our guest in this issue is Andrew J. Viterbi. Born in Bergamo, Italy, on 9 March 1935, he obtained the S.B. and S.M. degrees (jointly in 1957) from the Massachusetts Institute of Technology and the Ph.D. degree (1962) from University of Southern California, all in electrical engineering. He spent equal portions of his career in industry and academia. In industry, he cofounded Linkabit Corporation (1968) and Qualcomm Inc., of which he became vice chair and chief technical officer (1985–2000). In academia, Dr. Viterbi was a professor at University of California in Los Angeles (1963–1973) and then a professor at University of California in San Diego (1975–1994), at which he is now a Professor Emeritus. Since 2000 he has been the president of the Viterbi Group, LLC, a technical and investment company. His research has focused on aspects of digital communication. Most recently, he concentrated his efforts on establishing CDMA as the multiple access technology of choice for cellular telephony and wireless data communication. Dr. Viterbi authored the books *Principles of Coherent Communication* (1966) and *CDMA: Principles of Spread Spectrum Communication* (1995), and coauthored the book *Principles of Digital Communication and Coding* (1979). He received numerous awards, including the IEEE Alexander Graham Bell award (1984), the IEEE Claude Shannon award (1991), the Franklin Medal (2005), and six honorary doctorates. He received an honorary title from the president of Italy (2001) and served on the U.S. president's information technology advisory committee (1997–2001).

Andrew Viterbi considers that proving the error bound of the convolutional code has been his happiest professional moment. He appreciates originality and consistency in his collaborators, with some of whom (e.g., Jack Wolf and Chuck Wheatley) he goes back 40 years. In turn, they call him (affectionately) Andy. When he is not busy at work or planning philanthropic actions, the most recent of which endowed the Viterbi School of Engineering at the University of Southern California in Los Angeles, Andrew Viterbi enjoys reading, particularly history books. He believes that our paths go "per aspera ad astra" (translated from Latin as "through hardships to the stars"). In this issue, our guest tells the story of his principal research contribution, the Viterbi algorithm, which is used in most digital cellular phones and digital satellite receivers, as well as in such diverse fields as magnetic recording, voice recognition, and DNA sequence analysis.

—Adriana Dumitras, George Moschytz
"DSP History" column editors
adrianad@ieee.org, moschytz@isi.ee.ethz.ch

HOW IT ALL BEGAN

My first job at the Jet Propulsion Laboratories (JPL) in 1957 was in analog signal processing, specifically dealing with a coherent tracking circuit known as the phase locked loop. (This circuit has since been often digitized with its key component, the voltage controlled oscillator (VCO), implemented as a digital frequency synthesizer; however, this occurred long after my involvement.) The same device could also be used as a demodulator for an analog modulated signal. Digital modulation was not prevalent at the time. However, by 1960 I had been exposed to the elegance of the statistical communication and information theories as a student at the Massachusetts Institute of Technology. This motivated my struggle to understand how digital modulation could improve the performance of space communication, JPL's primary role for NASA. The resulting paper [1] was just a limiting case of Shannon's coding theorem for the white Gaussian channel. The demodulator consisted of multiple correlators for orthogonal and biorthogonal signals and could be implemented by analog circuitry. Already then, a digital implementation seemed more practical, especially after my colleague Rick Green developed an efficient implementation, which was dubbed the "Green machine" but turned out to be the Fast Hadamard Transform. While decoding had previously been considered to be *data* processing, because it dealt only with binary symbols, this system was an early implementation of digital *signal* processing because the demodulator returned real number values, which after quantization were processed by the decoder. Most subsequent digital communication receivers likewise realized combinations of demodulators-decoders using DSP.

1053-5888/05/$20.00©2005IEEE

THE VITERBI ALGORITHM

Moving to an academic position at University of California in Los Angeles in 1963, I struggled to teach information theory, in particular convolutional codes and sequential decoding, that were then considered to achieve the closest possible performance to the Shannon limit. Mixing teaching and research, I found a way to prove the superiority of convolutional codes over block codes for a given degree of decoding complexity. A key step in the proof was the development of a new nonsequential decoding algorithm [2], which later was labeled with my name. Though the algorithm actually sprang from my desire to simplify and clarify the material taught in an information theory course, the underlying stimulus for my interest in the research was the improvement of deep space communication efficiency. This was an area in which I had worked for nearly a decade and appeared to be an ideal application, since the deep space communication channel was the closest to the theoretical model of the additive Gaussian channel. It took the broader and longer-term view of academic research (in those times adequately supported by governmental R&D agencies) to look beyond the immediate project requirements and attempt a fresh approach.

As it was recognized by the early 1970s, the algorithm was a maximum likelihood decision device for any symbol sequence that could be modeled as a Markov chain [3]. Many phenomena in the physical world, as well as in computational abstractions, can be modeled as Markov sequences and described by Markov graphs. The Viterbi algorithm is a computationally efficient technique for determining the most probable path taken through a Markov graph. The graph (and underlying Markov sequence) is characterized by a finite set of states $\{S_0, S, \ldots, S_n\}$, state transition probabilities $\Pr(S_j \rightarrow S_i)$ and the output (observable parameter) probabilities $p(y|S_j \rightarrow S_i)$ for all $0 \leq i, j \leq n$, where n is the cardinality of the state space and the observables y are either discrete or continuous random variables. An example of a four-state Markov graph is illustrated in Figure 1, where only the nonzero probability transitions are shown. Thus, for example, from state S_1 the only nonzero transition probabilities are those to states S_2 and S_3, while from state S_3 they are those to itself, S_3, and to S_2.

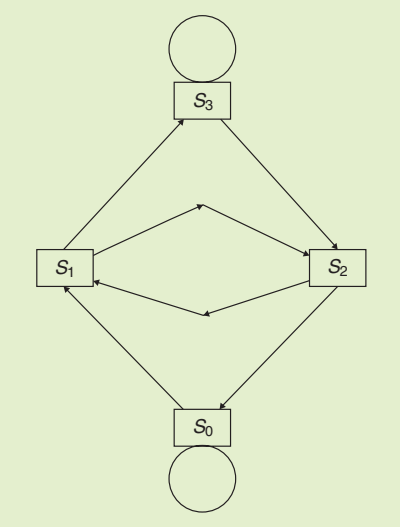
It is convenient for the description of the algorithm to view the multistep evolution of the path through the graph by means of a multistage replication of the Markov graph known as a *trellis* diagram. An example is shown in Figure 2, which illustrates the trellis diagram corresponding to the four-state Markov graph in Figure 1.

In Figure 2, if we label each branch (allowable transition between states) of the trellis diagram by its branch metric $m[]$, and each state at each node level by its state metric $M[]$, where $m[]$ and $M[]$ are defined in "The Viterbi Algorithm" sidebar, the state metrics at node level K are obtained from the state metrics at the level $K-1$ by a) adding to each state metric at level $K-1$ the branch metrics that connect it to states at the Kth level and b) preserving for each state at level K only the largest sum that arrives to it. In addition, if at each level we delete all branches other than the branch that produces this maximum, only one path will remain through the trellis. This path leads from the origin to each state at the Kth level and is the most probable path reaching it from the origin. In typical (though not all) applications, both the initial state (origin) and the final state (end) are selected to be S_0. Therefore, the algorithm produces the most probable path through the trellis both starting and ending at S_0. The Viterbi algorithm is summarized in "The Viterbi Algorithm" sidebar.

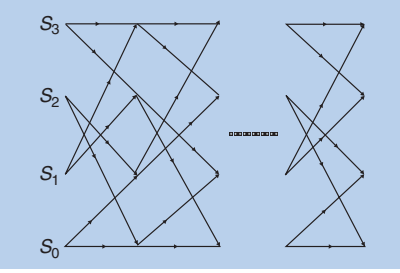
AN EARLY APPLICATION

Numerous applications of this algorithm have appeared over the past several decades. The earliest application, for which the algorithm was originally proposed [2], was the maximum likelihood decoding of convolutionally coded digital sequences transmitted over a noisy channel. However, note that the algorithm was proposed not so much to develop an efficient maximum likelihood decoder for a convolutional code but primarily to establish bounds on its error correcting performance. Currently, this application of the algorithm forms an integral part of the majority of wireless telecommunication systems, incorporated in both satellite digital television receivers and cellular mobile handsets.

The combination of a convolutional encoder and channel is shown in Figure 3. In the simplest case, one bit at a time enters the L-stage shift register and the n linear combiners. Each combiner is a modulo-2 adder of the contents of some subset of the L shift register stages and generates n binary symbols. These symbols are transmitted serially. For example, they are transmitted as binary amplitude ($x = +1$ or -1) modulation of a carrier signal. At the receiver, the demodulator generates an output y, which is either a real number or the result of quantizing the real number to one of a finite set of values. The conditional densities $p(y \mid x)$ of the channel outputs are assumed to be mutually independent, meaning that the channel is considered a "memoryless channel." A common example is the additive white Gaussian noise (AWGN) channel for which each y is the



[FIG1] A 4-state Markov graph example.



[FIG2] Trellis diagram for the Markov graph of Figure 1.

THE VITERBI ALGORITHM

The Viterbi algorithm is a computationally efficient technique for determining the most probable path taken through a Markov graph. The algorithm is most easily described using the trellis diagram representation corresponding to the graph (for instance, the trellis diagram in Figure 2).

Let us denote by $y(1), y(2), \ldots, y(k), \ldots$ the successive observables (not illustrated) in the diagram, each of which may be a vector corresponding to multiple observations per branch. Let us denote by $y(k)$ the observable(s) for the kth successive branch. Let $S(k)$ be any state at the kth successive node level. We shall remove the subscripts until necessary.

The goal is to find the most probable path through the trellis diagram. A fundamental assumption is that successive Markov state probabilities $Pr[S(k-1) \rightarrow S(k)]$ are mutually independent for all k, as are the conditional output probabilities $p[y(k)|S(k-1) \rightarrow S(k)]$.

For any given path from the origin $(k = 0)$ to an arbitrary node $(k = K)$ and the states $S(0), S(1), \ldots, S(K)$, using the index $0 \leq k \leq K$ the relative path probability (likelihood function) is given by

$$L = \prod_{k=1}^{K} \Pr[S(k-1) \rightarrow S(k)]\, p[y(k)|S(k-1) \rightarrow S(k)]. \quad (1)$$

For computational purposes it is more convenient to consider the logarithm of L:

$$ln(L) = \sum_{k=1}^{k} m[y(k); S(k-1), S(k)], \quad (2)$$

where we define the *branch metric* between any two states at the $(k-1)$th and kth node levels as

$$m[y(k); S(k-1), S(k)] = ln\{\Pr[S(k-1) \rightarrow S(k)]\} + ln\{p[y(k)|S(k-1) \rightarrow S(k)]\}. \quad (3)$$

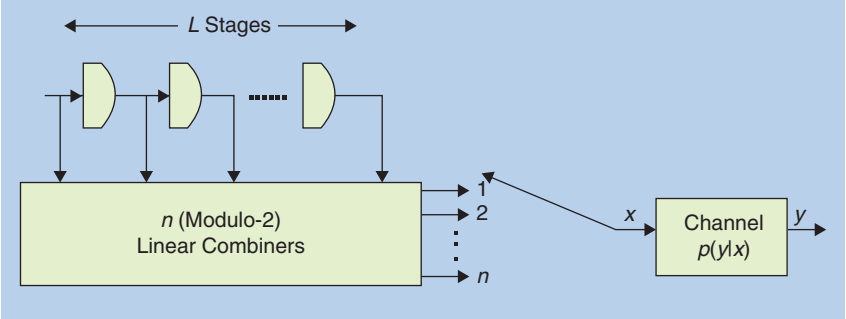
Let us define the *state metric*, $M_K(S_i)$ of the state $S_i(K)$ as the maximum over all paths leading from the origin to the ith state (i.e., state S_i) at the Kth node level (while inserting again subscripts where necessary):

$$M_K(S_i) = \text{Max}\left\{\sum_{k=1}^{k-1} m[y(k); S(k-1), S(k)] + m[y(K); S(K-1), S_i(K)]\right\}. \quad (4)$$

All paths $S(0), S(1), \ldots, S(K-1)$.

To maximize the above sum over K terms, it is sufficient to maximize the sum over the first $K-1$ terms for each state $S_j(K-1)$ at the $(K-1)$ th node and then maximize the sum of this and the Kth term over all states S $(K-1)$. This gives the recursion that is the core of the Viterbi algorithm:

$$M_K(S_i) = \text{Max}\{M_{K-1}(S_i) + m[y(K);\ S_j(K-1),\ S_i(K)]\} \quad (5)$$
$$S_j(K-1).$$



[FIG3] Convolutional encoder and channel.

sum of the encoded symbol x and a Gaussian random noise variable, with all noise variables mutually independent. This channel model is closely approximated by satellite and space communication applications and, with appropriate caution, it can also be applied to terrestrial communication design.

The communication system model just described gives rise naturally to the Markov graph representation. The 2^L states correspond to the states of the contents of the L stage register. Thus S_0 corresponds to the contents being all zeros, S_1 to the first stage containing a one and all the rest zeros, etc. Since only one input bit changes each time, each state has only two branches both exiting and entering it, each from two other states. One exiting branch corresponds to a zero value entering the register, while the other exiting branch corresponds to a one value. Figure 1 could be used to represent a two-stage encoder, with the state indices being the decimal equivalents of the binary register contents. It is generally assumed that all input bits are equally likely to be zero or one, so the state transition probabilities, $P(S_j \rightarrow S_i) = 1/2$ for each branch. Hence, the first term of the branch metric m in (3) can be omitted since it is the same for each branch. As for the second term of m, the conditional probability density $p(y|S_j \rightarrow S_i)$ is equal to $p(y|x)$, where x is an n-dimensional binary vector generated by the n modulo-two adders for each new input bit. This vector x corresponds to the encoder output from a state transition; y is the random vector corresponding to the n noise-corrupted outputs for the n channel inputs represented by the vector x. For the AWGN, $\ln p(y|x)$ is proportional to the inner product of the two vectors x and y.

The convolutional encoder and its Markov graph just described represent a rate $1/n$ code, since each input bit generates n output symbols. To generalize to any rational rate $(m/n) < 1$, m input bits enter each time and the register shifts in blocks of m. The Markov graph changes only in having each state connected to 2^m other states. Another generalization is to map each binary vector x, not into a vector of n binary values, +1 or –1, but into a

(continued on page 142)

constellation of points in two or more dimensions. A common case is that of quadrature amplitude modulation (QAM). For example, for $n = 4$, 16 points may be mapped into a two-dimensional grid and the value in each dimension modulates the amplitude of one of the two quadrature components of the sinusoidal carrier. Here x is the two-dimensional vector representing one of the 16 modulating values, and y is the corresponding demodulated channel output. Multiple generalizations of this approach abound in the literature and in real applications. In most cases, this multidimensional approach is used to conserve bandwidth at the cost of a higher channel signal-to-noise requirement.

BECOMING UBIQUITOUS

As was recognized by the early 1970s, the algorithm was a maximum likelihood decision device for any symbol sequence that could be modeled as a Markov chain [3]. Early on this made it a candidate for channel equalizers and magnetically recorded data. Later it was applied (by others) to a much broader set of applications, which could be represented by so-called hidden Markov models (HMMs). These ranged from speech recognition to DNA sequence analysis.

Considered hopelessly complex in 1967 when it was published [2], the Viterbi algorithm rode the curve of Moore's law, from a rack of equipment in 1975 to a fraction of a tiny chip today. Convolutional codes with Viterbi decoding first used in NASA spacecraft and military satellites in the 1970s ultimately became the workhorse of commercial voice and data communication satellites and direct broadcast satellites, residing in tens of millions of receivers today. The application returned to earth in the 1990s as second generation digital mobile phones became adopted. All international standards for second- and third-generation cellular voice phones employ convolutional codes and Viterbi decoding, reaching over a billion users. All this has been a pleasant surprise to me. Like my contemporaries of the 1960s, I had not foreseen the impact that integrated circuits would have on future systems.

In 1990 at an IEEE Communication Workshop I gave a talk titled "From Proof to Product," which recounted the strange origin of a DSP technique which, thanks to semiconductor integration exponentially progressing, went from a theoretical curiosity to a well-accepted system component in a quarter century and then to a component of a ubiquitous product in another decade.

REFERENCES

[1] A.J. Viterbi, "On coded phase coherent communication," *IRE Trans. Space Electron. Telemetry*, vol. SET-7 pp. 3–14, 1961.

[2] A.J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory,* vol. IT-13, pp. 260–269, 1967.

[3] G.D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268–278, 1973. SP

propagation, are not neglected. Additionally, as previously noted, the limitations and practical aspects of wireless communications are frequently highlighted throughout the text.

The depth of the discussion varies throughout the book. The text begins with a good introduction to radio propagation, slow (shadowing) and fast fading, and ray tracing and includes a number of empirical path-loss models, including the dielectric canyon model with ten rays. Following the discussion on propagation, and before discussing classical digital modulation techniques and their performance, the author discusses the capacity of wireless channels. The high-level information theory sections are followed by detailed discussions of capacity with diversity, capacity of frequency-selective fading and time varying channels, and capacity with varying degrees of side information. (The author could have devoted an appendix to the elements of information theory.) There are also sufficient references for readers seeking more details. Throughout the book, pertinent references are provided and these are particularly useful in sections that are discussed at high level, such as the overview on synchronization and carrier phase recovery in Chapter 5. One area where the book could be improved is that of equalization, the discussion of which would have required more depth. In particular, the development of the z-transform approach to linear equalizers does not make it sufficiently clear that the analysis applies to infinite length equalizers. In the general landscape of the book, this could be a possible improvement in an otherwise generally lucid presentation of the important topics in wireless communications.

In terms of aesthetics and functionality, the formatting, font, and figures are all well laid out and organized, making the book easy to read. The figures are generally informative. As an example, Figure 3.8 nicely summarizes the combined effects of path loss, shadowing, and narrowband fading. (Interestingly, in a course the author teaches at Stanford, five of the 19 lectures are devoted to channel models and various fading phenomena.)

In perspective, on a scale of engineering detail and theory, the book would be positioned between Tse and Viswanath's book and the book *Wireless Communications—Principles and Practice* by T.S. Rappaport. The former book has a more narrow focus, and goes into considerable detail on multi-input, multi-output systems. The latter book contains a more detailed discussion on various current wireless systems and standards and less on equalization, diversity and coding. Rappaport includes a discussion on speech coding, which is not included in Goldsmith's book. Overall, *Wireless Communications* by Andrea Goldsmith is an excellent, reader-friendly book, which maintains the high standards of the Cambridge University Press series initiated in 1998. SP