Explanation of the Viterbi Algorithm

USA

You're running a campaign to seek funding for educational computers from Sacramento, California to Augusta, Maine. You need to convince each state to adopt a legal measure that would do so. You know you aren't going to win every state, so you want to visit the states that are most likely to consider your measure. You (unlike me) decide to plan out the trip so that it will be most efficient. You start thinking. From Sacramento, there are a number of capital cities (Phoenix, AZ; Carson City, NV; Salem, OR). You number each possibility and assign a likelihood to each.

From each of your first cities, there are a number of second cities that you can drive to. You number each secondary city and assign an additional likelihood. Immediately you realize that both from Salem, OR and from Carson City, NV you can drive to Boise, ID. You don't need to keep both paths (Sacramento-Carson City-Boise and Sacramento-Salem-Boise); you can eliminate one of them and continue on. So, you pick the one with the highest overall likelihood of adoption and move on.

You continue to eliminate possibilities in this manner. By the time you arrive in Augusta, you'll probably have a handful of remaining paths you could take to get from Sacramento to Augusta. You can then decide which of the paths is best.

South America

You're running a campaign to seek funding for educational computers from Santiago, Chile to Cayenne, French Guiana. You need to convince each state to adopt a legal measure that would do so. You know you aren't going to win every state, so you want to visit the states that are most likely to consider your measure. You (unlike me) decide to plan out the trip so that it will be most efficient. You start thinking. From Santiago, there are a number of capital cities (Buenos Aires, Argentina; Montevido, Uruguai; Asuncion, Paraguay; La Paz, Bolivia; and Lima, Peru). You number each possibility and assign a likelihood to each.

From each of your first cities, there are a number of second cities that you can drive to. You number each secondary city and assign an additional likelihood. Immediately you realize that both from La Paz, Bolivia and Asuncion, Paraguay you can drive to Brasillia, Brazil. You don't need to keep both paths (Santiago-La Paz-Brasilia and Santiago-Asuncion-Brasilia); you can eliminate one of them and continue on. So, you pick the one with the highest overall likelihood of adoption and move on.

You continue to eliminate possibilities in this manner. By the time you arrive in Cayenne, French Guiana, you'll probably have a handful of remaining paths you could take to get from Santiago to Cayenne. You can then decide which of the paths is best.

India

You're running a campaign to seek funding for educational computers from Thiruvananthapuram, Kerala to Shimia, Himachal Pradesh. You need to convince each state to adopt a legal measure that would do so. You know you aren't going to win every state, so you want to visit the states that are most likely to consider your measure. You (unlike me) decide to plan out the trip so that it will be most efficient. You start thinking. From Thiruvananthapuram, there are two capital cities (Chennai, Tamil Nadu; and Bangalore, Karnataka). You number each possibility and assign a likelihood to each.

From each of your first cities, there are a number of second cities that you can drive to. You number each secondary city and assign an additional likelihood. Immediately you realize that both from Bangalore and Chennai you can drive to Hyderabad, Andhra Pradesh. You don't need to keep both paths (Thiruvananthapuram-Bangalore-Hyderabad and Thiruvananthapuram-Chennai-Hyderabad); you can eliminate one of them and continue on. So, you pick the one with the highest overall likelihood of adoption and move on.

You continue to eliminate possibilities in this manner. By the time you arrive in Shimia, you'll probably have a handful of remaining paths you could take to get from Thiruvananthapuram to Shimia. You can then decide which of the paths is best.

China

You're running a campaign to seek funding for educational computers from Lhasa, Xizang to Harbin, Heilongjiang. You need to convince each state to adopt a legal measure that would do so. You know you aren't going to win every state, so you want to visit the states that are most likely to consider your measure. You (unlike me) decide to plan out the trip so that it will be most efficient. You start thinking. From Lhasa, there are a number of capital cities (Urumqi, Xinjiang; Xining, Qinghai; Chengdu, Sichuan; and Kunming, Yunnan). You number each possibility and assign a likelihood to each.

From each of your first cities, there are a number of second cities that you can drive to. You number each secondary city and assign an additional likelihood. Immediately you realize that both from Xining, Qinghai and from Chengdu, Sichuan you can drive to Lanzhou, Gansu. You don't need to keep both paths (Lhasa-Xining-Lanzhou and Lhasa-Chengdu-Lanzhou); you can eliminate one of them and continue on. So, you pick the one with the highest overall likelihood of adoption and move on.

You continue to eliminate possibilities in this manner. By the time you arrive in Harbin, you'll probably have a handful of remaining paths you could take to get from Lhasa to Harbin. You can then decide which of the paths is best.

Europe

You're running a campaign to seek funding for educational computers from Lisbon, Portugal to Moscow, Russia. You need to convince each state to adopt a legal measure that would do so. You know you aren't going to win every state, so you want to visit the states that are most likely to consider your measure. You (unlike me) decide to plan out the trip so that it will be most efficient. You start thinking. You decide to go to Madrid, Spain and then Andorra, Andorra. However, from Andorra, there are a number of capital cities (Monaco, Monaco; Bern, Switzerland; Paris France; Dublin, Ireland; and London, UK). You number each possibility and assign a likelihood to each.

From each of your first cities, there are a number of second cities that you can drive to. You number each secondary city and assign an additional likelihood. Immediately you realize that both from Paris, France and from Bern, Switzerland, you can travel to Luxembourg, Luxembourg. You don't need to keep both paths (Andorra-Paris-Luxembourg and Andorra-Bern-Luxembourg); you can eliminate one of them and continue on. So, you pick the one with the highest overall likelihood of adoption and move on.

You continue to eliminate possibilities in this manner. By the time you arrive in Moscow, you'll probably have a handful of remaining paths you could take to get from Lisbon to Moscow. You can then decide which of the paths is best.

Generalization

This process of elimination is exactly how the Viterbi Algorithm works. When there are two paths that converge to the same point, the path with the highest likelihood is chosen.

To save myself some typing substitute the following:

Continent/Country	City0	City1a	City1b	City2a
USA	Sacramento	Carson City	Salem	Boise
South America	Santiago	La Paz	Asuncion	Brasilia
India	Thiruvananthapuram	Bangalore	Chennai	Hyderabad
China	Lhasa	Xining	Chengdu	Lanzhou
Europe	Andorra	Paris	Bern	Luxembourg

More generally, the possible 1st-stop cities (City1) will be called City1a, City1b, City1c, etc. The possibly 2nd-stop cities (City2) will be called City2a, City2b, City2c, etc. The possible 3rd-stop cities (City3) will be called City3a, City3b, City3c, etc. And so on.

Memory | Dependency

There's an additional feature of the above analogy that I should point out. When we considered travelling from city to city, we assumed that the likelihood of adoption showed no memory (i.e. it was independent at each city). Thos that don't want to be can confused can resolve this complication by just taking my word that each city represents a state (no, not a political state—a state vector). It isn't, for example, a single bit in a transmitter/receiver, but a sequence of bits. Alternatively, one can go through the rest of this section to resolve this complicated dependency in a more detailed manner.

Let's now suppose that when you arrive in City2a (Boise|Brasilia|Hyderabad|Lanzhou|Luxmebourg), word will have spread on whether you were successful in City1a (Carson City|La Paz|Bangalore|Ximing|Paris). So, the likelihood of educational computer adoption at City2a is not just based on City2a; it depends as well on the likelihood of adoption at City1a or City1b.

Let's assume that in each city, the politicians only commiserate with their neighbors. So, the likelihood of adoption at each city visited only depends on whether their neighbor adopted the computers. (The alternative is for the cities to commiserate with not just their neighboring cities but with their neighbors and their neighbor's neighbors.) When I generalize to receivers, this will be described as K=2.

No worries: we just have to now modify our criterion for eliminating possibilities. Instead of two paths landing at the same city, we need two paths to have the same current city and last-visited city. Then, we can eliminate one of the paths. So, we can't compare the path City0-City1a-City2a to City0-City1b-City2a and eliminate one, because the likelihood at City2a depends on City1a and City1b. We have two possible likelihoods for City2a and that duplicity screws up our decision at City3. What I'm basically saying is that the likelihood assigned to City2 is different if we follow the path City0-City1a-City2a versus if we follow the path City0-City1b-City2a. Once we get to City3, we *can* decide between City0-City1a-City2a-City3b and City0-City1b-City2a-City3b; we just can't eliminate either path until we get to City3.

Maximum-Likelihood Receivers

An application of the Viterbi algorithm is in a maximum-likelihood receiver.

The Viterbi decoder is used in receivers to search for the set of bits b[n] that's closest to the received signal r[n]. Knowing that the bits go through a transmit filter, a channel (modeled as another filter), and a receive filter, the real minimization is done on b[n]*h[k]-r[n], where h[k] is the cumulative effect of all the filters (transmit, channel, and receive); * represents convolution. Essentially, the Viterbi decoder does a search, hypothesizing every possible sequence of bits b[n] that were sent and computing what they would look like when they are received: b[n]*h[k]. These hypothetical received signals are then compared to the actual received sequence.

One requirement/assumption of the Viterbi decoder is that h[k] needs to be a FIR filter. Let's assume it of length K. As with our cross-country tour of the USA, if two (or more) paths end up giving the same K previous bits, we can eliminate one.

The upshot is that if you receive a sequence of samples r[n] and you know they originated from a bit sequence b[n] that then goes through some set of filters h[k], the Viterbi detector searches for the "right" sequence b[n] that minimizes the error $\sum|b[n]*h[k] - r[n]|^2$.

I will apply this amazing feat of the Viterbi detector to Sigma-Delta DAC's in a future article.

About

Poojan Wagh is a contributing author at Circuit Sage. He has worked in the areas of direct-digital-RF PA/DAC's, CMOS GSM/EDGE transceivers, WCDMA continuous-time CMOS ADC's, CMOS TV tuners, and software-definable radio front-ends. He also runs his own circuit design blog at http://www.circuitdesign.info/blog.