AIRR Handout on Viterbi and Kalman Filters

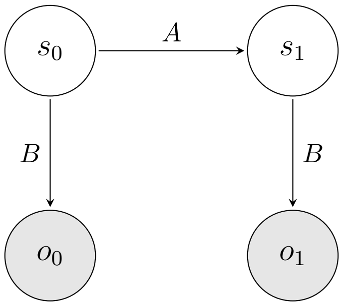
Nishanth Kumar

Last Revised: 02/27/2025

1 Viterbi Algorithm

The Viterbi algorithm is really just max-product in discrete HMM's! Let's look at a concrete example and follow the notation from lecture.

Consider the below HMM. Assume both states s_0 and s_1 are binary.



Here, the transition probability matrix A is given by.

$A =$

$s_t \backslash s_{t+1}$	T	F
T	0.5	0.5
F	0	1

Emission probabilities are given by the matrix B:

$B =$

	$o = T$	$o = F$
T	0.5	0.5
F	0	1

The initial probabilities are uniform:

$$\pi(T) = 0.5, \quad \pi(F) = 0.5.$$

We observe the sequence $(o_0, o_1) = (F, T)$ and seek the most likely sequence (s_0^*, s_1^*) using the Viterbi algorithm.

1.1 Step 1: Initialization

At $t = 0$, we compute:

$$\delta_0(T) = \pi(T) \cdot B_T(F) = 0.5 \times 0.5 = 0.25.$$

$$\delta_0(F) = \pi(F) \cdot B_F(F) = 0.5 \times 1 = 0.5.$$

1.2 Step 2: Forward Pass (Recursion)

At $t = 1$, given $o_1 = T$, we compute $\delta_1(T)$ and $\delta_1(F)$.

1.2.1 Computing $\delta_1(T)$

$$\delta_1(T) = \max_{s_0} [\delta_0(s_0) \cdot P(T \mid s_0)] \cdot P(T \mid T).$$

- From $s_0 = T$:

$$\delta_0(T) \cdot P(T \mid T) = 0.25 \times 0.5 = 0.125.$$

- From $s_0 = F$:

$$\delta_0(F) \cdot P(T \mid F) = 0.5 \times 0 = 0.$$

Thus,

$$\delta_1(T) = \max(0.125, 0) \times 0.5 = 0.0625.$$

Best previous state:

$$\psi_1(T) = T.$$

1.2.2 Computing $\delta_1(F)$

$$\delta_1(F) = \max_{s_0} [\delta_0(s_0) \cdot P(F \mid s_0)] \cdot P(T \mid F).$$

- From $s_0 = T$:

$$\delta_0(T) \cdot P(F \mid T) = 0.25 \times 0.5 = 0.125.$$

- From $s_0 = F$:

$$\delta_0(F) \cdot P(F \mid F) = 0.5 \times 1 = 0.5.$$

However, since $P(T \mid F) = 0$, we get:

$$\delta_1(F) = \max(0.125, 0.5) \times 0 = 0.$$

Best previous state does not matter because the probability is 0.

1.3 Step 3: Backtrace

$$s_1^* = \arg\max_{s} \delta_1(s).$$

Since

$$\delta_1(T) = 0.0625, \quad \delta_1(F) = 0,$$

we get

$$s_1^* = T.$$

Then, using $\psi_1(T)$:

$$s_0^* = \psi_1(s_1^*) = \psi_1(T) = T.$$

1.4 Final Result

The most likely state sequence is:

$$(s_0^*, s_1^*) = (T, T).$$

This should make sense — from the matrix A we see that if the state is ever F, it must stay F and there is no probability it becomes T. We also know from B that if a particular state is F, then we will observe it as F with probability 1. Thus, given we observe $o_1 = T$, we can conclude s_1 must have been T. Given this, we can conclude s_0 must have also been T because otherwise A would have made it impossible to transition to $s_1 = T$.

2 1D Robot Localization with a Kalman Filter

2.1 Problem Setup

We consider a robot moving along a straight line, estimating its position using a Kalman Filter. The robot moves forward by 1 meter per time step on average, but with some random noise. A sensor provides noisy position measurements.

Process Model The state at time t, denoted x_t, follows:

$$x_t = x_{t-1} + 1 + w_{t-1}, \tag{1}$$

where $w_{t-1} \sim \mathcal{N}(0, W)$ represents process noise, modeled as Gaussian with mean zero and variance W.

Measurement Model The sensor provides noisy measurements:

$$y_t = Hx_t + v_t, \tag{2}$$

where:

- H is the measurement matrix (in this case, a scalar),
- $v_t \sim \mathcal{N}(0, R)$ is measurement noise.

For this example, we assume:

- $W = 1$ (process noise variance),
- $R = 2$ (measurement noise variance),
- $H = 1$ (sensor directly observes position),
- Initial state estimate: $\hat{x}_{0|0} = 0$,
- Initial covariance: $Q_{0|0} = 10$.

Observations At each time step, the robot receives the following sensor readings:

$$y_1 = 0.5, \quad y_2 = 2.2, \quad y_3 = 3.5. \tag{3}$$

We now apply the Kalman Filter step by step.

2.2 Kalman Filter Equations

At each time step t, the filter follows:

Prediction Step

$$\hat{x}_{t|t-1} = \hat{x}_{t-1|t-1} + 1, \tag{4}$$

$$Q_{t|t-1} = Q_{t-1|t-1} + W. \tag{5}$$

Measurement Update Step

$$K_t = \frac{Q_{t|t-1}H}{HQ_{t|t-1}H + R}, \tag{6}$$

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(y_t - H\hat{x}_{t|t-1}), \tag{7}$$

$$Q_{t|t} = (1 - K_tH)Q_{t|t-1}. \tag{8}$$

Since $H = 1$ in this case, these simplify to:

$$K_t = \frac{Q_{t|t-1}}{Q_{t|t-1} + R}, \tag{9}$$

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(y_t - \hat{x}_{t|t-1}), \tag{10}$$

$$Q_{t|t} = (1 - K_t)Q_{t|t-1}. \tag{11}$$

2.3 Numerical Example

Let's walk through computing the first three time steps.

2.3.1 Time Step 1 ($t = 1$)

Prediction

$$\hat{x}_{1|0} = \hat{x}_0 + 1 = 0 + 1 = 1, \tag{12}$$
$$Q_{1|0} = Q_{0|0} + W = 10 + 1 = 11. \tag{13}$$

Measurement Update (Given $y_1 = 0.5$)

$$K_1 = \frac{11}{11+2} = \frac{11}{13} \approx 0.846, \tag{14}$$
$$\hat{x}_{1|1} = 1 + 0.846(0.5 - 1) = 1 - 0.423 = 0.577, \tag{15}$$
$$Q_{1|1} = (1 - 0.846) \times 11 = 1.69. \tag{16}$$

2.3.2 Time Step 2 ($t = 2$)

Prediction

$$\hat{x}_{2|1} = \hat{x}_{1|1} + 1 = 0.577 + 1 = 1.577, \tag{17}$$
$$Q_{2|1} = Q_{1|1} + W = 1.69 + 1 = 2.69. \tag{18}$$

Measurement Update (Given $y_2 = 2.2$)

$$K_2 = \frac{2.69}{2.69+2} = \frac{2.69}{4.69} \approx 0.573, \tag{19}$$
$$\hat{x}_{2|2} = 1.577 + 0.573(2.2 - 1.577) = 1.934, \tag{20}$$
$$Q_{2|2} = (1 - 0.573) \times 2.69 = 1.149. \tag{21}$$

2.3.3 Time Step 3 ($t = 3$)

Prediction

$$\hat{x}_{3|2} = \hat{x}_{2|2} + 1 = 1.934 + 1 = 2.934, \tag{22}$$
$$Q_{3|2} = Q_{2|2} + W = 1.149 + 1 = 2.149. \tag{23}$$

Measurement Update (Given $y_3 = 3.5$)

$$K_3 = \frac{2.149}{2.149+2} = \frac{2.149}{4.149} \approx 0.518, \tag{24}$$
$$\hat{x}_{3|3} = 2.934 + 0.518(3.5 - 2.934) = 3.227, \tag{25}$$
$$Q_{3|3} = (1 - 0.518) \times 2.149 = 1.036. \tag{26}$$

2.4 Discussion and Observations

- The **covariance decreases over time**, reflecting increased confidence in the estimate.
- The Kalman Gain K_t balances the prediction vs. the measurement. As $Q_{t|t}$ decreases, the filter trusts its own prediction more.
- The estimated positions are pulled towards the expected motion despite noisy observations.