Data Association: Multi-model Framework

$$
\mathbf{z}_k = \begin{array}{cc} & N_k \rightarrow \\ N_t \downarrow & \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array} = \mathbf{z}_{k,1} + \mathbf{z}_{k,2} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}
$$

$$
\mathbf{z}_k = \mathbf{z}_{k,1} + \mathbf{z}_{k,2} + \ldots + \mathbf{z}_{k,M}
$$

$$
z_{km}(t,i) = \begin{cases} 1; & \text{if observation } y_{k,i} \text{ originated from and falls in validation} \\ & \text{gate of target t} \\ 0; & \text{otherwise} \end{cases}
$$

State process $\Phi_k = (\Phi_{1,k}, \Phi_{2,k}, \ldots, \Phi_{N_t,k})^T$ Model state $\phi_k^m(t)$

Observation process $\mathbf{y}_k = (y_{k,1}, y_{k,2}, \ldots, y_{k,N_k})^T$

MAP State Estimation

Bayes' rule

$$p(\Phi_k|\mathbf{Y}^k) = \boxed{\frac{p(\mathbf{Y}_k|\Phi_k)}{p(\mathbf{Y}_k|\mathbf{Y}^{k-1})}p(\Phi_k|\mathbf{Y}^{k-1})}$$

$$p(\mathbf{Y}_k|\Phi_k) = \int_{\mathbf{Z}_k} p(\mathbf{Y}_k, \mathbf{Z}_k|\Phi_k)d\mathbf{Z}_k = \int_{\mathbf{Z}_k} p(\mathbf{Y}_k|\mathbf{Z}_k, \Phi_k)p(\mathbf{Z}_k|\Phi_k)d\mathbf{Z}_k$$

With incomplete data

$$p(\Phi_k|\mathbf{X}^k) = \frac{p(\mathbf{X}_k|\Phi_k)}{p(\mathbf{X}_k|\mathbf{X}^{k-1})}p(\Phi_k|\mathbf{X}^{k-1})$$

MAP estimate of the state

EM Algorithm

E-step

$$Q(\Phi_k|\hat{\Phi}_k^{(p)}) = E\{\log p(\mathbf{X}_k|\Phi_k)|\mathbf{Y}_k, \hat{\Phi}_k^{(p)}\}$$

M-step

$$\hat{\Phi}_k^{(p+1)} = \arg\max_{\Phi_k}\left[Q(\Phi_k|\hat{\Phi}_k^{(p)}) + \log p(\Phi_k|\hat{\Phi}_{k-1})\right]$$

Neural Network for Data Association

- Solve problem similar to Salesman Traveling Problem
  - Finding optimal path given distance among cities
  - Optimize path using cost function

A B C D E F

A B C D E F

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Rows and Columns represent
Targets and Observations
Each entry is neuron
In our case, neuron outputs
assignment weight

Hopfield network – dynamic equations and energy function

Dynamic equations for neural network $V_i(t) = g(u_i(t))$

$$
\begin{aligned}
\frac{du_i(t)}{dk} &= -\frac{u_i(t)}{k_0} - A\sum_{\substack{s=1\\ s\neq t}}^{N_t} V_i(s) - B\sum_{\substack{j=0\\ j\neq i}}^{n_k} V_j(t) - C\left(\sum_{j=0}^{N_k} V_j(t) - 1\right) \\
&- [D + E(N_t - 1)V_i(t) + (D+E)\rho_i(t) + E\left(N_t - 1 - \sum_{s=1}^{N_t}\rho_i(s)\right)
\end{aligned}
$$

Neural energy function

$$
\mathcal{E}' = -\frac{1}{2}\sum_{i=1}^{N_k}\sum_{j=1}^{N_k}\sum_{t=1}^{N_t}\sum_{s=1}^{N_t} T_{ij}(ts)V_i(t)V_j(s) - \sum_{i=1}^{N_k}\sum_{t=1}^{N_t} V_i(t)I_i(t)
$$

$$
I_i(t) = C + (D+E)\rho_i(t) + E(N_t - 1 - \sum_{s=1}^{N_t}\rho_i(s))
$$

$$
T_{ij}(ts) = -[A\delta_{ij}(1-\delta_{ts} + B\delta_{ts}(1-\delta_{ij}) + C\delta_{ts} + D\delta_{ts}\delta_{ij} + E(N_t - 1)\delta_{ts}\delta_{ij}]
$$

$$
\delta_{ij} = \begin{cases} 1, & if \quad i = j \\ 0, & if \quad i \neq j \end{cases}
$$

Hopfield network implementation

Approximate difference equation for neural energy function

$$u_i^{(p+1)}(t) = \left(\frac{s_0-\zeta}{s_0}\right)u_i^{(p)}(t) - \zeta A\sum_{\substack{s=1\\s\neq i}}^{N_t} V_i^{(p)}(s) - \zeta B\sum_{\substack{j=0\\j\neq i}}^{N_k} V_j^{(p)}(t) - \zeta C\left(\sum_{j=0}^{N_k} V_j^{(p)}(t) - 1\right)$$

$$- \zeta[D + E(N_t - 1)]V_i^{(p)}(t) + \zeta(D+E)\rho_i(t) + \zeta E\left(N_t - 1 - \sum_{s=1}^{N_t}\rho_i(s)\right)$$

Neuron input voltage

$$u_i(t) = \frac{u_0}{2}\ln\left(\frac{V_i(t)}{1-V_i(t)}\right)$$

Neuron output voltage

$$V_i(t) = g(u_i(t)) = \frac{1}{2}\left(1 + \tanh\frac{u_i(t)}{u_0}\right)$$

Initial neuron input voltage $V_i(t) = \rho_i(t)$ or $V_i(t) = \dfrac{1}{N_k+1}$

Constraint on neuron input voltage $\sum_{i=1}^{N_k} V_i(t) = 1$

Energy function for Tracking algorithm

- Energy function

$$
\begin{aligned}
\mathcal{E} \;=\;& \frac{A}{2}\sum_{i=l}^{N_k}\sum_{t=1}^{N_t}\sum_{\substack{s=1\\ s\neq t}}^{N_t} V_i(t)V_i(s) + \frac{B}{2}\sum_{i=1}^{N_k}\sum_{t=1}^{N_t}\sum_{\substack{j=1\\ j\neq i}}^{N_k} V_i(t)V_j(t) + \frac{C}{2}\sum_{t=1}^{N_t}\left(\sum_{i=1}^{N_k} V_i(t) - 1\right)^2 \\
+\;& \frac{D}{2}\sum_{i=1}^{N_k}\sum_{t=1}^{N_t}(V_i(t)-\rho_i(t))^2 + \frac{E}{2}\sum_{i=1}^{N_k}\sum_{t=1}^{N_t}\sum_{\substack{s=1\\ s\neq t}}^{N_t}\left(V_i(t) - \sum_{\substack{j=1\\ j\neq i}}^{N_k}\rho_i(s)\right)^2
\end{aligned}
$$

- Normalized likelihood

$$
\rho_i(t) = \frac{\sum_{m=1}^{M} p_m[y_{k,i}|\mathbf{z}_{k,i} = e_t, \phi_k^{m(p)}(t)]\mu_k^m(t)}{\sum_{j=1}^{N_k}\sum_{m=1}^{M} p_m[y_{k,j}|\mathbf{z}_{k,j} = e_t, \phi_k^{m(p)}(t)]\mu_k^m(t)}
$$

Problem Statement: Target Tracking

- Target Dynamics - Constant Velocity
- Target State – Position and Velocity
- Observation - Position
- Estimation of hidden state parameters