Topics in Brain Computer Interfaces
CS295-7

Professor: Michael Black
TA: Frank Wood

Spring 2005

Kalman Filtering
\[ \tilde{z}_t = H \tilde{x}_t + \text{noise} \]

Approximation: Linear Gaussian (generative) model

observation model

\[ \tilde{z}_{t-j} \sim \mathcal{N}(H \tilde{x}_t, Q_t) \]

Full covariance \( Q \) matrix models correlations between cells.

\( H \) models how firing rates relate to full kinematic model (position, velocity, and acceleration).
Bayesian Inference

\[ p(\text{kinematics} \mid \text{firing}) = \frac{p(\text{firing} \mid \text{kinematics}) p(\text{kinematics})}{p(\text{firing})} \]

Posterior

\text{likelihood} (evidence)

\text{prior (a priori - before the evidence)}

\text{a posteriori probability} (after the evidence)

normalization constant (independent of mouth)

We \textit{infer} hand kinematics from uncertain evidence and our prior knowledge of how hands move.
Multi-Modal Likelihood

How can we represent this?
Non-Parametric Approximation

- We could sample at regular intervals

**Problems?**

- most samples have low probability – wasted computation
- How finely to discretize
- High dimensional space – discretization impractical
Factored Sampling

Weighted samples

$S = \{(x^{(i)}, w^{(i)}); \ i = 1...N\}$

Normalized likelihood:

$$W^{(n)}_t = \frac{p(z_t|x^{(n)}_t)}{\sum_{i=1}^N p(z_t|x^{(i)}_t)}$$
Monte-Carlo Sampling

Given a weighted sample set

\[ S = \{(x^{(i)}, w^{(i)}); i = 1...N\} \]
Bayesian Tracking

Posterior over model parameters given an image sequence.

\[ p(x_t | Z_t) = \int \kappa p(z_t | x_t) (p(x_t | x_{t-1}) p(x_{t-1} | Z_{t-1})) dx_{t-1} \]

Temporal model (prior)

Likelihood of observing the firing rates given the hand kinematics.

Posterior from previous time instant

Monte Carlo integration
Particle Filter

Posterior $p(x_{t-1} | Z_{t-1})$

Isard & Blake '96
Particle Filter

Posterior \( p(x_{t-1} | Z_{t-1}) \)

Isard & Blake ‘96
Particle Filter

Posterior $p(x_{t-1} \mid Z_{t-1})$

Temporal dynamics $p(x_t \mid x_{t-1})$

Isard & Blake ‘96
Particle Filter

Posterior $p(x_{t-1} \mid Z_{t-1})$

Temporal dynamics $p(x_t \mid x_{t-1})$

Likelihood $p(z_t \mid x_t)$

Isard & Blake ‘96
Particle Filter

Posterior \( p(x_{t-1} \mid Z_{t-1}) \)

Temporal dynamics
\( p(x_t \mid x_{t-1}) \)

Likelihood
\( p(z_t \mid x_t) \)

normalize

Posterior
\( p(x_t \mid Z_t) \)

Isard & Blake '96
Pseudocode

condense1step

% generate cumulative distribution for posterior at t-1

....

% generate a vector of uniform random numbers.
% if a the number is greater than refreshRate then
  % generate a vector of uniform random numbers
  % use these to search the cumulative probability
  % find the indices of the corresponding particles
  % for each of these particles, predict the new state (eg. Add Gaussian noise!)
  % for each of these new states compute the log likelihood
% else generate a particle at random and compute its log likelihood.
% find the maximum log likelihood and subtract it from all the other log likelihoods
% construct the posterior at time t by exponentiating all the log likelihoods and normalizing so they sum to 1.
Linear Gaussian Likelihood

*Generative model* for the observation:

\[ z_k = H_k x_k + q_k \]

\[ H_k \in \mathbb{R}^{c \times d}, \quad Q_k \in \mathbb{R}^{c \times c}, \quad q_k \sim N(0, Q_k), \quad k = 1, 2, \ldots, M. \]
Explicit Form

The likelihood model is equivalent to that

$$z_k \sim N(H_k x_k, Q_k)$$

The conditional probability has explicit form:

$$p(z_k | x_k) = \frac{1}{((2\pi)^c \det(Q_k))^{1/2}} \exp\left(-\frac{1}{2} (z_k - H_k x_k)^T Q_k^{-1} (z_k - H_k x_k)\right)$$
Linear Gaussian Temporal Prior

Temporal prior of the state:

\[ x_k = A_k \ x_{k-1} + w_k \]

\( A_k \in \mathbb{R}^{d \times d}, \quad w_k \sim N(0, W_k), \quad W_k \in \mathbb{R}^{d \times d}, \quad k = 2,3,\ldots M. \)
Explicit Form

The prior model is equivalent to that

\[ x_{k+1} \sim \mathcal{N}(A_k x_k, W_k) \]

The conditional probability has explicit form:

\[
p(x_{k+1}|x_k) = \frac{1}{((2\pi)^d \det(W_k))^{1/2}} \exp\left(-\frac{1}{2} (x_{k+1} - A_k x_k)^T W_k^{-1} (x_{k+1} - A_k x_k)\right)
\]
Kalman Filter Model

Definition:

System Equation:
\[ x_k = A_k x_{k-1} + w_k, \quad w_k \in N(0, W_k) \]

Measurement Equation:
\[ z_k = H_k x_k + q_k, \quad q_k \in N(0, Q_k) \]

Assumption:
All random variables have Gaussian distributions and they are linearly related
Linear Gaussian Model

\[ p(x_t \mid Z_t) = \kappa \, p(z_t \mid x_t) \int (p(x_t \mid x_{t-1}) \, p(x_{t-1} \mid Z_{t-1})) \, dx_{t-1} \]

Some basic facts about Gaussians:
- Marginal of a Gaussian is Gaussian
- Gaussian times a Gaussian is Gaussian

\[ p_1(x) = N(\mu_1, \Sigma_1), \quad p_2(x) = N(\mu_2, \Sigma_2) \]

\[ p_1(x)p_2(x) = zN(\mu_3, \Sigma_3) \]

\[ \mu_3 = \Sigma_3 \left( \Sigma_1^{-1} \mu_1 + \Sigma_2^{-1} \mu_2 \right) \]

\[ \Sigma_3 = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1} \]
Linear Gaussian Model

\[ p(x_t \mid Z_t) = \kappa \, p(z_t \mid x_t) \int (p(x_t \mid x_{t-1}) \, p(x_{t-1} \mid Z_{t-1})) \, dx_{t-1} \]

A linear transformation of a Gaussian distributed random variable is also Gaussian:

\[
\begin{align*}
x & \sim N(\bar{x}, Q) \\
y & = Ax + b \\
y & \sim N(A\bar{x} + b, AQA^T)
\end{align*}
\]
Linear Gaussian Model

\[ p(x_t \mid Z_t) = \kappa p(z_t \mid x_t) \int (p(x_t \mid x_{t-1}) p(x_{t-1} \mid Z_{t-1})) \, dx_{t-1} \]

\[
\begin{align*}
N(Ax_{t-1}, W) & \quad N(\bar{x}_{t-1}, P_{t-1}) \\
\end{align*}
\]

\[ x_t \sim N(A\bar{x}_{t-1}, AP_{t-1}A^T + W) \]

\[ x_t \sim N(\hat{x}_t^-, P_t^-) \]
Linear Gaussian Model

\[
p(x_t \mid Z_t) = \kappa \ p(z_t \mid x_t) \int (p(x_t \mid x_{t-1}) p(x_{t-1} \mid Z_{t-1})) \, dx_{t-1}
\]

\[
N(Hx_t, Q)
\]

\[
x_t \sim N(\hat{x}_t^-, P^-)
\]

\[
const \times \exp \left( -\frac{1}{2} (z_t - Hx_t)^T Q^{-1} (z_t - Hx_t) - \frac{1}{2} (x_t - \hat{x}_t^-)^T P_t^{-1} (x_t - \hat{x}_t^-) \right)
\]
Linear Gaussian Model

\[ p(x_t \mid z_t) \]

\[ = \text{const} \times \exp \left( -\frac{1}{2} (z_t - Hx_t)^T Q^{-1} (z_t - Hx_t) - \frac{1}{2} (x_t - \hat{x}_t)^T P_t^{-1} (x_t - \hat{x}_t) \right) \]

\[ = N(\hat{x}_t \mid P_t) \]

\[ \hat{x}_t = \left( P_t^{-1} + H^T Q^{-1} H \right)^{-1} \left( P_t^{-1} \hat{x}_t + H^T Q^{-1} z_t \right) \]

\[ \hat{x}_t = P_t \left( P_t^{-1} \hat{x}_t + H^T Q^{-1} z_t \right) \]

\[ P_t = \left( P_t^{-1} + H^T Q^{-1} H \right)^{-1} \]
Linear Gaussian Model

\[
\hat{x}_t = \left(P_t^{-1} + H^T Q^{-1} H\right)^{-1} \left(P_t^{-1} \hat{x}^- + H^T Q^{-1} z_t\right)
\]

\[
P_t = \left(P_t^{-1} + H^T Q^{-1} H\right)^{-1}
\]

Some algebra.

\[
\hat{x}_t = \hat{x}^- + K_t (z_t - H\hat{x}^-)
\]

\[
K_t = P_t H^T Q^{-1}
\]

\[
K_t = \left(P_t^{-1} + H^T Q^{-1} H\right)^{-1} H^T Q^{-1}
\]
Simplifying

Matrix inversion lemma

\[ K_t = \left( P_t^{-1} + H^T Q^{-1} H \right)^{-1} H^T Q^{-1} \]

\[ \left( P_t^{-1} + H^T Q^{-1} H \right)^{-1} \]

\[ = P_t^{-1} - P_t^{-1} H^T (Q^{-1} + HP_t^{-1} H^T)^{-1} HP_t^{-1} \]

\[ = P_t \]
**Kalman Filter Algorithm**

**Time Update**

Prior estimate

\[ \hat{x}_t^-= A \hat{x}_{t-1} \]

Error covariance

\[ P_t^- = AP_{t-1} A^T + W \]

Initial estimate of \( \hat{x}_{t-1} \) and \( P_{t-1} \)

**Measurement Update**

Posterior estimate

\[ \hat{x}_t = \hat{x}_t^- + K_t (z_t - H \hat{x}_t^-) \]

Error covariance

\[ P_t = P_t^- - P_t^- H^T (Q^{-1} + H P_t^- H^T)^{-1} H P_t^- \]

Kalman gain

\[ K_t = P_t H^T Q^{-1} \]

*Welch and Bishop 2002*
Learning Kalman Model

• In practice, the parameters in the model need to be estimated from training data. (In training data, we know both hidden states and measurements.)

• Common simplification: $A_k, H_k, W_k, Q_k$ are constant over time (independent of $k$).

• The $A, H, W, Q$ can be estimated by maximizing the joint probability $p(X_M, Z_M)$. 
\[ x_k = A_k x_{k-1} + w_k \]

\[ z_k = H_k x_k + q_k \]

\[ p(X_M, Z_M) = p(X_M) p(Z_M | X_M) \]

\[ = [p(x_1) \prod_{k=2}^{M} p(x_k | x_{k-1})] \prod_{k=1}^{M} p(z_k | x_k) \]
Splitting the Joint Distribution

\[
\begin{align*}
\arg\max_{A,W,H,Q} p(X_M, Z_M) &= \arg\max_{A,W} p(X_M) \arg\max_{H,Q} p(Z_M | X_M) \\
&= \arg\min_{A,W} f(A,W) \arg\min_{H,Q} g(H,Q)
\end{align*}
\]

where

\[
f(A, W) = -\alpha \log p(X_M) = \sum_{k=2}^{M} [ \log(\det W) + (x_k - Ax_{k-1})^T W^{-1} (x_k - Ax_{k-1}) ],
\]

\[
g(H, Q) = -\beta \log p(Z_M | X_M) = \sum_{k=1}^{M} [ \log(\det Q) + (z_k - Hx_k)^T Q^{-1} (z_k - Hx_k) ].
\]

How to optimize functions with matrix variables ???
Closed-form Solutions:

\[
A = \left( \sum_{k=2}^{M} x_k x_k^T \right) \left( \sum_{k=2}^{M} x_{k-1} x_{k-1}^T \right)^{-1},
\]

\[
W = \frac{1}{M-1} \left( \sum_{k=2}^{M} x_k x_k^T - A \sum_{k=2}^{M} x_{k-1} x_k^T \right),
\]

\[
H = \left( \sum_{k=1}^{M} z_k z_k^T \right) \left( \sum_{k=1}^{M} x_k x_k^T \right)^{-1},
\]

\[
Q = \frac{1}{M} \left( \sum_{k=1}^{M} z_k z_k^T - H \sum_{k=1}^{M} x_k z_k^T \right).
\]
Off-line Reconstruction

69 cells with 1.5 minutes of training data

- Actual hand position
- Estimated/decoded position (reconstruction)
Accuracy

Continuous 2D hand motion (off-line reconstruction):

<table>
<thead>
<tr>
<th>Method</th>
<th>RMSE (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population vector</td>
<td>8.66</td>
</tr>
<tr>
<td>Linear regression method</td>
<td>2.55</td>
</tr>
<tr>
<td>Kalman filter</td>
<td>2.18</td>
</tr>
</tbody>
</table>

As number of cells increases:

![Graphs showing MSE and CorrCoeff against number of cells]
Optimal “Lag”

Measurement Equation

\[ \tilde{z}_k = H \tilde{x}_k + \tilde{q}_k \]

Firing precedes motion:

* Uniform: lag \( j \) time steps (1 time step = 70ms)

\[ \tilde{z}_{k-j} = H \tilde{x}_k + \tilde{q}_k \quad j = 0,1,2,3,4 \]

* Non-uniform: lag \((j_1, j_2, \cdots, j_{42})\) time steps
## Reconstruction and Lag

<table>
<thead>
<tr>
<th>Methods</th>
<th>CC $(x, y)$</th>
<th>MSE $(cm^2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kalman (0ms lag)</td>
<td>(0.77, 0.91)</td>
<td>6.96</td>
</tr>
<tr>
<td>Kalman (70ms lag)</td>
<td>(0.79, 0.93)</td>
<td>6.67</td>
</tr>
<tr>
<td><strong>Kalman (140ms lag)</strong></td>
<td><strong>(0.81, 0.93)</strong></td>
<td><strong>6.09</strong></td>
</tr>
<tr>
<td>Kalman (210ms lag)</td>
<td>(0.81, 0.89)</td>
<td>6.98</td>
</tr>
<tr>
<td>Kalman (280ms lag)</td>
<td>(0.76, 0.82)</td>
<td>8.91</td>
</tr>
<tr>
<td>Kalman (non-uniform)</td>
<td>(0.82, 0.93)</td>
<td>5.24</td>
</tr>
</tbody>
</table>
RECONSTRUCTION (TEST DATA)

reconstructed
true

\begin{align*}
\text{x-position} & \quad 20 & \quad 15 & \quad 10 & \quad 5 & \quad 0 & \quad 5 & \quad 10 & \quad 15 & \quad 20 \\
\text{y-position} & \quad 10 & \quad 5 & \quad 0 & \quad 5 & \quad 10 & \quad 15 & \quad 20 \\
\text{x-velocity} & \quad 2 & \quad 1 & \quad 0.5 & \quad 1 & \quad 2 & \quad 5 & \quad 10 & \quad 15 & \quad 20 \\
\text{y-velocity} & \quad 2 & \quad 1 & \quad 0.5 & \quad 1 & \quad 2 & \quad 5 & \quad 10 & \quad 15 & \quad 20 \\
\text{x-acceleration} & \quad 1 & \quad 0.5 & \quad 1 & \quad 2 & \quad 5 & \quad 10 & \quad 15 & \quad 20 \\
\text{y-acceleration} & \quad 1 & \quad 0.5 & \quad 1 & \quad 2 & \quad 5 & \quad 10 & \quad 15 & \quad 20
\end{align*}
Covariance

Diagonal covariance assumes noise in firing rates is statistically independent.

\[
p(z_t \mid x_t) = \prod_{i=1}^{C} p(z_{i,t} \mid x_t) = \prod_{i=1}^{C} \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{1}{2\sigma_i^2}(z_{i,t} - H_i x_t)^2\right)
\]

Kalman decoding

* Full covariance: MSE = 5.99 cm\(^2\)
* Diagonal covariance: MSE = 6.35 cm\(^2\)
## Off-line Decoding

<table>
<thead>
<tr>
<th># of cells</th>
<th>Kalman filter</th>
<th>Linear regression</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$CC$</td>
<td>$RMSE$ (cm)</td>
</tr>
<tr>
<td>23</td>
<td>((0.79, 0.82))</td>
<td>3.61</td>
</tr>
<tr>
<td>30</td>
<td>((0.88, 0.79))</td>
<td>3.26</td>
</tr>
<tr>
<td>36</td>
<td>((0.75, 0.74))</td>
<td>4.36</td>
</tr>
<tr>
<td>26</td>
<td>((0.71, 0.76))</td>
<td>4.48</td>
</tr>
<tr>
<td>69</td>
<td>((0.88, 0.89))</td>
<td>3.11</td>
</tr>
<tr>
<td>69</td>
<td>((0.86, 0.88))</td>
<td>3.26</td>
</tr>
</tbody>
</table>

**Kalman filter**

$CC = 0.81\pm0.06$, $RMSE = 3.7\pm0.6$ (cm)

**Linear regression**

$CC = 0.74\pm0.05$, $RMSE = 4.3\pm0.6$ (cm)
Decoding results over all six experiments:
On-line Neural Control

Neural control of a computer cursor in real time.

Brain substitutes for hand.

Kalman filter decoder.
Only 18 cells.

Directly exploits the generative encoding model.

Target

Visual feedback
# On-line Task Performance

<table>
<thead>
<tr>
<th># of cells</th>
<th>Kalman filter</th>
<th>Linear regression</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>time</td>
<td>targets</td>
</tr>
<tr>
<td>17</td>
<td>60sec</td>
<td>38</td>
</tr>
<tr>
<td>30</td>
<td>105sec</td>
<td>55</td>
</tr>
<tr>
<td>36</td>
<td>57sec</td>
<td>28</td>
</tr>
<tr>
<td>69</td>
<td>45sec</td>
<td>28</td>
</tr>
</tbody>
</table>

**Average results:**

- **Kalman filter**: 33.75 targets/min
- **Linear regression**: 22.67 targets/min

50% improvement
Non-Gaussian Likelihood

Inhomogeneous Poisson:

$$p(z \mid x) = \frac{1}{z!} (Hx)^z e^{-Hx}$$

$Hx$ is the predicted mean firing rate.
$z$ is the observed rate for a single cell.

Problems? $Hx$ may be negative.
No clear way to model correlated noise.