Neural Decoding (II)

Neural Engineering
Pin Ball Task

Task: Hit random targets on the screen.

Motions: fast, unconstrained

Data (4.5 minutes):
• Position (Velocity, Acceleration)
• 1.5 minutes needed for training
• Firing rate (42 cells, non-overlapping 70ms bins)

Source: Brown University
For a prosthesis, this must be automated.

F. Wood, M. Black, C. Vargas-Irwin, M. Fellows, J. Donoghue

http://www0.cs.ucl.ac.uk/staff/c.archambeau/ATML/atml_files/fw08_neural_decoding.pdf
Encoding

Kettner et al ('88):

\[ z_k = b_0 + b_x x_k + b_y y_k \]  \hspace{1cm} \text{(Linear in position)}
We want to reconstruct hand position from firing rates.
Let's assume a linear relationship:

\[ x_k = f_x^T z_k + b_x \]

\[ x_k = \begin{bmatrix} f_{x,1} & \cdots & f_{x,n} \end{bmatrix} \begin{bmatrix} z_{k,1} \\ \vdots \\ z_{k,n} \end{bmatrix} + b_x = \begin{bmatrix} b_x & f_{x,1} & \cdots & f_{x,n} \end{bmatrix} \begin{bmatrix} 1 \\ z_{k,1} \\ \vdots \\ z_{k,n} \end{bmatrix} \]
Linear filtering

Solve for \( f_x \)

\[
\begin{align*}
x_k &= [1 \ z_k^T \ z_{k-1}^T \ \cdots \ z_{k-20}^T] f_x \\
x_{k-1} &= [1 \ z_{k-1}^T \ z_{k-2}^T \ \cdots \ z_{k-21}^T] f_x \\
x_{k-2} &= [1 \ z_{k-2}^T \ z_{k-3}^T \ \cdots \ z_{k-22}^T] f_x \\
&\vdots \\
x &= Z f_x
\end{align*}
\]
Solve for $f_x$

\[ x = Z f_x \]

\[ f_x = (Z^T Z)^{-1} Z^T x \]

\[ f_x = \text{pinv}(Z)x \]

Reminder: this is least squares regression.
Linear decoding

Fit $f_x$ given training data.

Decoding is now simple.

Given a vector of observed firing activity

$$Z_k^T = \begin{bmatrix} 1 & z_{1,k} & \cdots & z_{n,k} & z_{1,k-1} & \cdots & z_{n,k-1} & \cdots & z_{n,k-20} \end{bmatrix}$$

“project” through the filter:

$$x_k = Z_k^T f_x$$
Neural Responses and Linear Fit
Decoding (off-line)

**x-position**

**y-position**

*True Reconstruction*
Linear Filter

\[ x_k = \mathbf{f} \cdot \mathbf{Z}_k + a \]

- learned “filter”
- constant offset
- hand position
- vector of firing rates for 42 cells over 20 bins (1.4 sec)

Simple regression model, fast decoding, reasonable reconstruction

No explicitly probabilistic model, No uncertainty estimation, slow encoding

https://slideplayer.com/slide/4738308/
Decoding by Kalman Filter

Inferring Hand Motion from Multi-Cell Recordings in Motor Cortex using a Kalman Filter

Wei Wu*, Michael Black †, Yun Gao*, Elie Bienenstock* §, Mijail Serruya §, Ammar Shaikhouni §, Carlos Vargas †§, John Donoghue §

Brown University

*Applied Mathematics †Computer Science §Neuroscience

https://slideplayer.com/slide/4738308/
What is an **Kalman Filter**?
It is an iterative mathematical process to quickly estimate the true value, position, velocity...

\[ T_i \text{(temp)} \]

Initial estimate ± (error)

\[ x = \text{measured temp} \]

\[ t \text{ (time)} \]

estimate temp using Kalman Filter

(actual temp)
Kalman Filter For Dummies

http://bilgin.esme.org/BitsAndBytes/KalmanFilterforDummies

Kalman Decoding

\[ Y_{k+1}^{\text{corrected}} = Y_{k+1}^{\text{predicted}} + k(X_k^{\text{measured}} - Y_k^h) \]

Time Update (prediction)

1. Project the state ahead
   \[ \hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \]
2. Project the error covariance ahead
   \[ P_k^- = AP_{k-1}A^T + Q \]

Measurement Update (correction)

1. Compute the Kalman Gain
   \[ K_k = P_k^-H^T(HP_k^-H^T + R)^{-1} \]
2. Update the error covariance
   \[ P_k = (I - K_kH)P_k^- \]

Initial estimates at \( k = 0 \)

The outputs at \( k \) will be the input for \( k+1 \)

Spike Counts

Linear Encoding Model
Kalman Filtering

Enter a priori estimate $\hat{X}_k$ and its error covariance $\hat{P}_k$.

Compute Kalman Gain

$K_k = \hat{H}_k^T (\hat{H}_k \hat{H}_k^T + R)^{-1}$

Project ahead

$\hat{x}_{k|k-1} = \Phi(t_{k-1}, t_k) \hat{x}_k$

$\hat{P}_{k|k-1} = \Phi(t_{k-1}, t_k) \hat{P}_k \Phi^T(t_{k-1}, t_k)$

Compute Error Covariance for Updated Estimate

$\hat{P}_k = (I - K_k \hat{H}_k) \hat{P}_k$

Update Estimate with Measurement $y_k$

$\tilde{x}_k = \hat{x}_k + K_k (y_k - \hat{H}_k \hat{x}_k)$

Kalman Decoding

$y_{corrected}^k = y_{predicted}^k + k(X_k^{measured} - Y_k h)$

Linear Encoding Model

Spike Counts

Counts Spike Encoding Linear Model
Neural Response and Kalman Fit

Fig. 3. Fitting multiple linear models: in the first two columns, the first row shows the empirical mean firing rate distributions of two cells with respect to hand position; the second row shows the corresponding linear fit to the data (Kalman filter model); the third row shows the empirical distribution of the sum of firing rates of these two cells and Kalman fit; the fourth row shows the two components recovered by the SKFM when applied the combined data. We see that these two linear components are very similar to the linear fits for the original two cells. The two right columns show the same behavior when we consider firing rate as a function of hand velocity. Note that the Kalman and SKF models used here fit linear models with respect to position, velocity and acceleration. The figure shows just the position or velocity component of these models.
Kalman Filter for Beginners

with MATLAB Examples

Phil Kim
Decoding by Particle Filter

Particle filter:
1. A method for approximating the probability density function by multiple samples
2. This is also called as Monte Carlo Filter, Bootstrap Filter, Sampling/Importance resampling (SIR) Filter.
3. Independently, Kitagawa and Gordon invented this.

Depending on observed data (we would predict that data have noise), we try to focus on expected point by some particles.

https://www.slideshare.net/HiroakiHamada/neural-decoding