Recursive Bayesian Decoding of Motor Cortical Signals by Particle Filtering

Brockwell, et al.

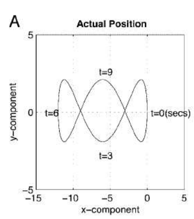
THERE WILL BE A QUIZ!

Setup

- 2 datasets
  - Simulation
  - Premotor cortex measurements
- 3 methods
  - PV
  - Linear Regression ("optimal linear estimation")
  - Particle Filter

Simulated Dataset

- Assume known path
- 60 realizations



- Tuning functions: (neuron *i=1…200*)

$$\lambda_i(v) = \max\,(k_i + m_i v \cdot d_i, 0)$$

  - Base firing rate *k*
  - Directional sensitivity *m*
  - Preferred direction (unit vector) *d*
    - Half on [0 $\frac{\pi}{2}$], half on [$\frac{\pi}{2}$ 2π] **nonuniform

Real Neuron Measurements

- "258 neurons in the subregion of ventral premotor cortex referred to by Gentilucci et al. (1998) as "region F4," collected individually in 258 separate experiments from four rhesus monkeys"

- It's described in Reina and Schwartz 03

Real Data Collection Setup

- Cube corners "center-out" task
- subdivide into 100 bins. (Extra credit. How can we stop doing this?)

- Ellipse task
- Subdivide each of 5 loops into 100 bins

- Both tasks: pretend all 258 measurements from one trial, use average velocity as ground truth

Population Vector

- Velocity at time t $\hat{v}_t^{(\mathrm{PV})} = \sum_{j=1}^{N} w_{t,j} d_j$
- N neurons
- Preferred directions d
- Weights w (for firing rates y)

$$w_{t,j} = (y_{t,j} - \bar{y}_j)/(y_j^{(\max)} - y_j^{(\min)})$$

Optimal Linear Estimation

- No assumption of uniformly distributed preferred directions
- Salinas and Abbott 94

Optimal Linear Estimation

- Chose the preferred directions using the known trajectory
- I.E. choose d to minimize

$$E[(v_t - \hat{v}_t)^T (v_t - \hat{v}_t)]$$

- Extra credit: then what did they use for the real data?
  - See pg 1901: preferred directions obtained using center-out data – but what did they do?

Particle Filter

Particle Filter (2500 particles)

- **State model**: random walk $v_t = v_{t-1} + \varepsilon_t$
- For error i.i.d ~N(0, 0.03 I)

- **Observation model**:

$$y_t^{(i)} | v_{t+\text{lag}_i} \sim \text{Poisson}(\lambda_i(v_{t+\text{lag}_i})) \quad i = 1, 2, \ldots, N$$

$$\lambda_i(v) = \exp(k_i + m_i v \cdot d_i + s_i \|v\|)$$

for nondirectional sensitivity s

Choosing PF parameters

- Preferred directions d estimated from center-out task
- Everything else from first 3 loops of ellipse task "using standard Poisson-family generalized linear models (McCullagh and Nelder 89)"

Lags

- Yowzers.

For each neuron, used lags yielding the best-fitting generalized linear model

Seems like there's a lot of art to this.

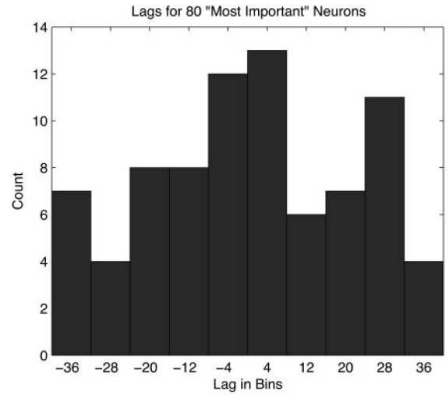
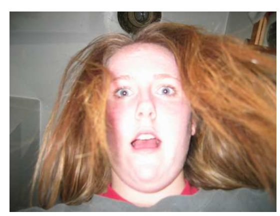


FIG. 1. Histogram of the lags, measured in time bins, of 80 neurons in the ventral premotor cortex data. Neurons were selected as those which spiked ≥10 times during the ellipse-tracing task and for which directional sensitivity was above the median directional sensitivity.

80 chosen by >10 spikes during first 5-loop trial, and having m(directional sensitivity) >.05

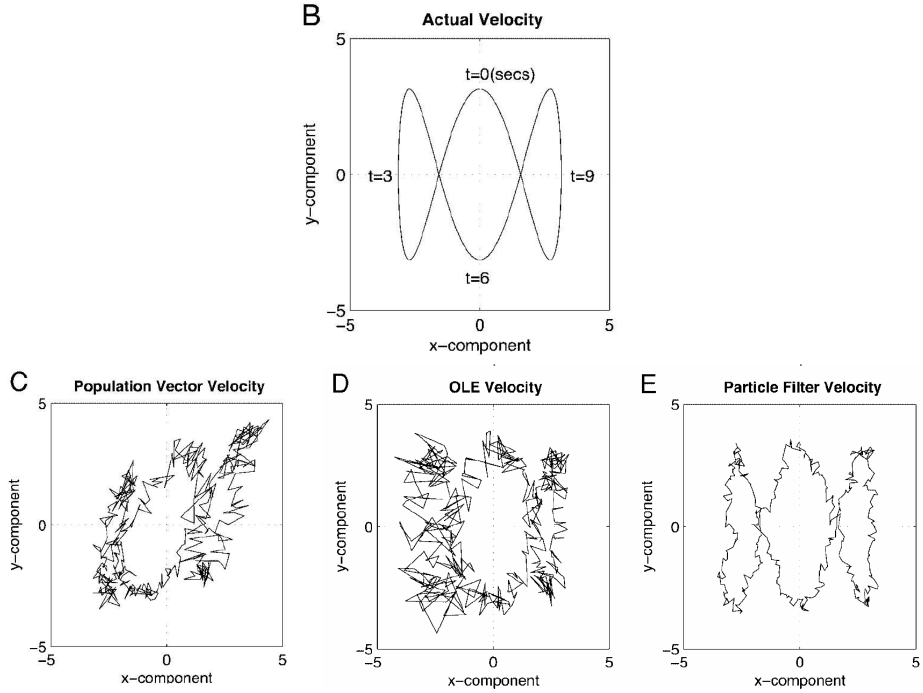
Results

- Drum roll…. PF wins!

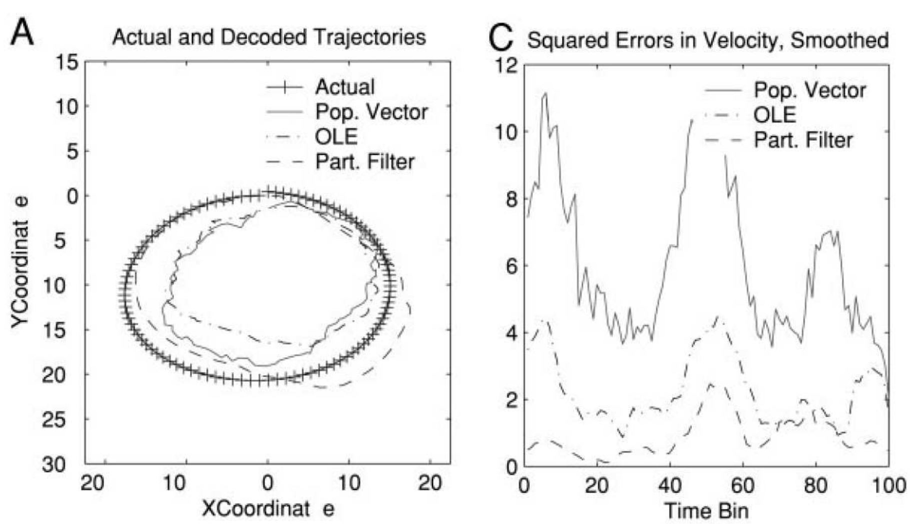


Shocked, shocked!

- For simulated data, MSE ~10 times smaller than PV and ~5 times smaller than OLE
- For real data, ~7 smaller than PV and ~3 times smaller than OLE



Real Neuron Results



Useful?

TABLE 2. *Decoding errors for ventral premotor cortex data summarized across time*

	PV	OLE	PF
ISE	6.245	2.362	0.886
MaxSE	33.978	9.349	4.904

ISE averages, while MaxSE maximizes, squared error across time. In terms of ISE, the PF offers a roughly sevenfold improvement over the PV algorithm.

10 Times Better MSE == 1/10 Neurons Required?

- Where does this come from?
- What assumptions implicit in this?

Particle Filter

PF is general method for conditional density propagation through time

We wish we had: $P(x_t \mid z_t)$

Bayes' Rule: $P(x_t \mid z_t) = \dfrac{P(z_t \mid x_t)P(x_t)}{P(z_t)}$

For time t, observations z_t, and state or value x_t

Particle Filter

PF is general method for conditional density propagation through time

We wish we had: $P(x_t \mid z_t)$

Bayes' Rule: $P(x_t \mid z_t) = \dfrac{P(z_t \mid x_t)P(x_t)}{yawn}$

For time t, observations z_t, and state or value x_t

Particle Filter

PF is general method for conditional density propagation through time

We wish we had: $P(x_t \mid z_t)$

Bayes' Rule: $P(x_t \mid z_t) = \dfrac{P(z_t \mid x_t)P(x_t \mid x_{t-1})}{yawn}$

For time t, observations z_t, and state or value x_t

Particle Filter

PF is general method for conditional density propagation through time

We wish we had: $P(x_t \mid z_t)$

Bayes' Rule: $P(x_t \mid z_t) \propto P(z_t \mid x_t)P(x_t \mid x_{t-1})$

For time t, observations z_t, and state or value x_t

Propagate a set of *samples* drawn from prob. density instead of parameterizing the density

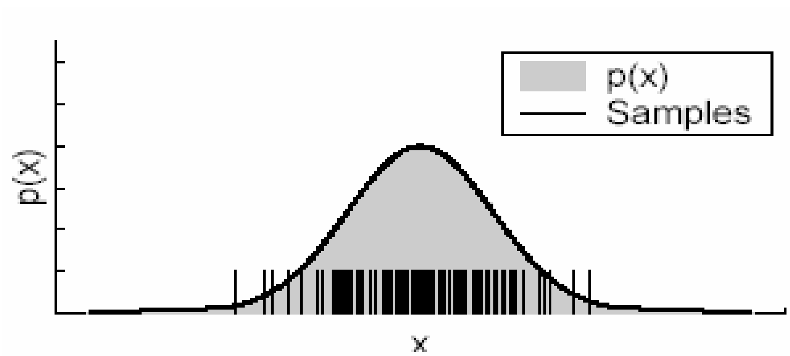


Figure: Dieter Fox

Dramatis Personae

Particle Set : $\chi_t = \left\langle x_t^m, w_t^m \right\rangle$

For mth particle m = 1 … M

'Kinetic' or 'State Transition' Model : $P(x_t^m \mid x_{t-1}^m)$

'Importance Weights' or 'Observation Model': $w_t^m = P(z_t \mid x_t^m)$

Posterior: $P(x_t \mid z_t) \propto P(z_t \mid x_t) P(x_t \mid x_{t-1})$

Use weights on samples from kinetic model density to approximate posterior density

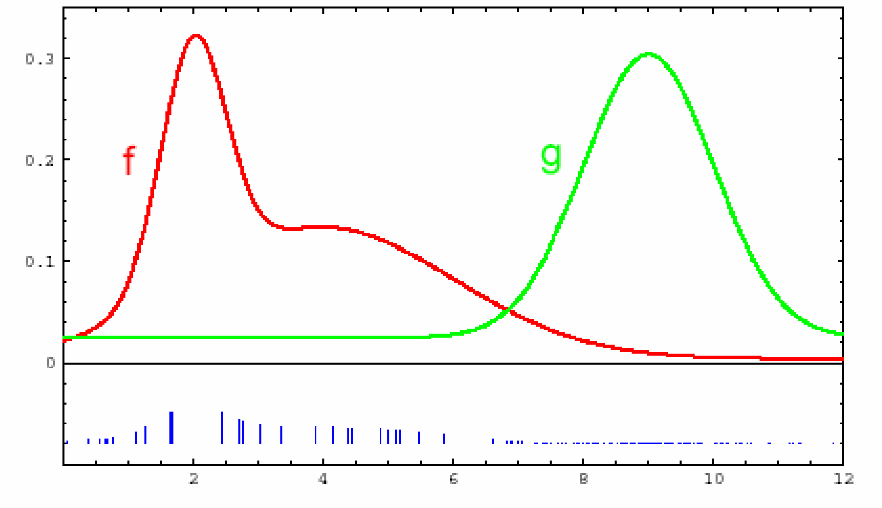


Figure: Dieter Fox

Particles

Red of tooth and claw.

Particle Filters – Resampling

Draw with replacement M particles from χ_t with probability w_t^m

Results in new particle set of the same size whose particles more closely represent the posterior

Likely to have duplicates, but that's OK. It's survival of the fittest!

(Like a lion)

Particle Filters – State Transition

- Apply state transition model to M surviving particles

$$P(x_t^m \mid x_{t-1}^m)$$

- Apply observation model

$$w_t^m = P(z_t \mid x_t^m)$$

- Rinse and repeat

- Reina and Schwartz make cool graphics