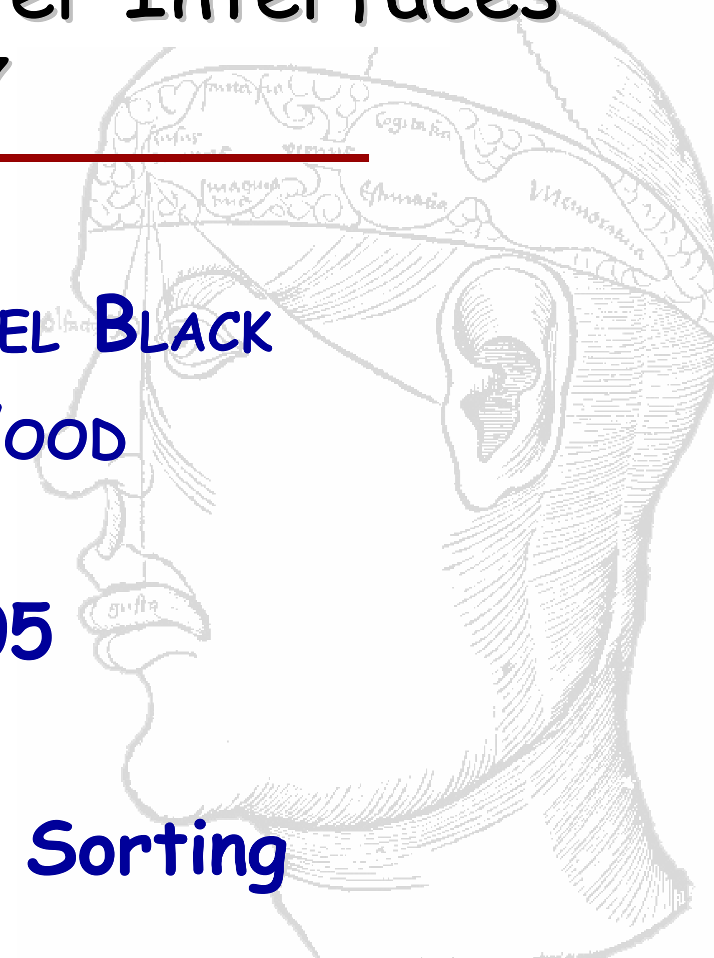
Topics in Brain Computer Interfaces
CS295-7



Professor: **Michael Black**

TA: **Frank Wood**

Spring 2005

Automated Spike Sorting

BROWN

Today

- Particle Filter Homework Discussion and Review
- Kalman Filter Review
- PCA Introduction
- EM Review
- Spike Sorting

BROWN

Particle Filtering Movies

Homework Results?

- Better than CC X .5, CC Y .8? How?
- What state estimator did you use (ML/E[])? Why?
- When did you estimate the state?
- Particle re-sampling schedule?
- Remaining questions?
- Initial state estimate?
- How did the homework synthesize with the lecture notes and readings?

BROWN

Viewing the Bayesian Recursion after implementing Particle Filtering

given $\mathrm{P}(\vec{x}_0)$ and $\mathrm{P}(\vec{z}_0)$ and the model $\mathrm{P}(\vec{z}_k | \vec{x}_k), \mathrm{P}(\vec{x}_k | \vec{x}_{k-1})$

$$\mathrm{P}(\vec{x}_0 \mid \vec{z}_0) = \frac{\mathrm{P}(\vec{z}_0 \mid \vec{x}_0)\mathrm{P}(\vec{x}_0)}{\mathrm{P}(\vec{z}_0)}$$

$$\mathrm{P}(\vec{x}_1 \mid \vec{z}_0) = \int \mathrm{P}(\vec{x}_1 \mid \vec{x}_0)\mathrm{P}(\vec{x}_0 \mid \vec{z}_0)\partial\vec{x}_0$$

$$\mathrm{P}(\vec{x}_1 \mid \vec{z}_0, \vec{z}_1) = \frac{\mathrm{P}(\vec{z}_1 \mid \vec{x}_1)\mathrm{P}(\vec{x}_1 \mid \vec{z}_0)}{\mathrm{P}(\vec{z}_1 \mid \vec{z}_0)}$$

$$\mathrm{P}(\vec{x}_2 \mid \vec{z}_0, \vec{z}_1) = \int \mathrm{P}(\vec{x}_2 \mid \vec{x}_1)\mathrm{P}(x_1 \mid \vec{z}_0, \vec{z}_1)\partial\vec{x}_1$$

$$\mathrm{P}(\vec{x}_2 \mid \vec{z}_0, \vec{z}_1, \vec{z}_2) = \frac{\mathrm{P}(\vec{z}_2 \mid \vec{x}_2)\mathrm{P}(\vec{x}_2 \mid \vec{z}_0, \vec{z}_1)}{\mathrm{P}(\vec{z}_2 \mid \vec{z}_0, \vec{z}_1)}$$

$$\vdots$$

System model

Start here with particles representing the posterior.

Observation model

Next Homework: The Kalman Filter

- Closed form solution to recursive Bayesian estimation where the observation and state models are linear + Gaussian noise.
- Seminal paper published in 1960:
  - R.E. Kalman, "A New Approach to Linear Filtering and Prediction Problems"

Observation model $$z_k = H_k \, x_k + q_k \qquad q_k \sim N(0, Q)$$

State model $$x_k = A_k \, x_{k-1} + w_k \qquad w_k \sim N(0, W)$$

BROWN

The Kalman Filter Algorithm

Time Update

Prior estimate

$$\hat{x}_k^- = A\hat{x}_{k-1}$$

Error covariance

$$P_k^- = AP_{k-1}A^T + W$$

Measurement Update

Posterior estimate

$$\hat{x}_k = \hat{x}_k^- + K_k(\vec{z}_k - H\hat{x}_k^-)$$

Error covariance

$$P_k = (I - K_k H)P_k^-$$

Kalman gain

$$K_k = P_k^- H^T (HP_k^- H^T + Q)^{-1}$$

Initial estimate of $\hat{x}_{k-1}$ and P_{k-1}

BROWN

Welch and Bishop 2002

Where do these equations come from?

- Find an unbiased minimum variance estimator of the state at time k+1 of the form

$$\hat{x}_{k+1} = K'_{k+1}\hat{x}_k + K_{k+1}z_{k+1}$$

We'll look at this today.

This bit is much trickier. A link to a full derivation is on the web.

- For $\hat{x}_{k+1}$ to be unbiased means:

$$E\left[\hat{x}_{k+1} - x_{k+1}\right] = 0$$

BROWN

Excerpted and modified from aticourses.com

½ Unbiased Estimate

Remember from the previous slide

$$E\left[\hat{x}_{k+1} - x_{k+1}\right] = 0$$
$$\hat{x}_{k+1} = K'_{k+1}\hat{x}_k + K_{k+1}z_{k+1}$$

Trick alert!

$$E\left[K'_{k+1}\hat{x}_k + K_{k+1}z_{k+1} - x_{k+1}\right] = 0$$

$$E\left[K'_{k+1}\hat{x}_k + K_{k+1}\left(Hx_{k+1} + q_{k+1}\right) - x_{k+1} - K'_{k+1}x_k + K'_{k+1}x_k\right] = 0$$

$$E\left[K'_{k+1}\left(\hat{x}_k - x_k\right) + K_{k+1}\left(H\left(Ax_k + w_{k+1}\right) + q_{k+1}\right) - \left(Ax_k + w_{k+1}\right) + K'_{k+1}x_k\right] = 0$$

$$E\left[K'_{k+1}\left(\hat{x}_k - x_k\right) + \left(K_{k+1}HA - A + K'_{k+1}\right)x_k + \left(K_{k+1}H - I\right)w_{k+1} + K_{k+1}q_{k+1}\right] = 0$$

$$\left(K_{k+1}HA - A + K'_{k+1}\right)E\left[x_k\right] = 0$$

$$\Rightarrow K_{k+1}HA - A + K'_{k+1} = 0$$

$$\text{or } K'_{k+1} = \left(I - K_{k+1}H\right)A$$

BROWN

Excerpted and modified from aticourses.com

Frank Wood - fwood@cs.brown.edu

Pulling it together (a bit)

Remember from the previous slide

$$K'_{k+1} = (I - K_{k+1}H)A$$

$$\hat{x}_{k+1} = K'_{k+1}\hat{x}_k + K_{k+1}z_{k+1}$$

$$\hat{x}_{k+1} = (I - K_{k+1}H)A\hat{x}_k + K_{k+1}z_{k+1}$$

$$\hat{x}_{k+1} = A\hat{x}_k + K_{k+1}(z_{k+1} - HA\hat{x}_k)$$

Posterior estimate

$$\hat{x}_k = \hat{x}_k^- + K_k(\vec{z}_k - H\hat{x}_k^-)$$

Error covariance

$$P_k = (I - K_k H)P_k^-$$

Kalman gain

$$K_k = P_k^- H^T (HP_k^- H^T + Q)^{-1}$$

Can get the Kalman gain by minimizing the variance of the estimation error.

BROWN

Excerpted and modified from aticourses.com

The Kalman Filter Algorithm

Time Update

Prior estimate

$$\hat{x}_k^- = A\hat{x}_{k-1}$$

Error covariance

$$P_k^- = AP_{k-1}A^T + W$$

Measurement Update

Posterior estimate

$$\hat{x}_k = \hat{x}_k^- + K_k(\vec{z}_k - H\hat{x}_k^-)$$

Error covariance

$$P_k = (I - K_k H)P_k^-$$

Kalman gain

$$K_k = P_k^- H^T (HP_k^- H^T + Q)^{-1}$$

Initial estimate of $\hat{x}_{k-1}$ and $P_{k\text{-}1}$

BROWN

Welch and Bishop 2002

Frank Wood - fwood@cs.brown.edu

Good time for a Break

- Changing gears to PCA/EM/Mixture Modeling

BROWN

Principal Component Analysis (PCA)

"The central idea of [PCA] is to reduce the dimensionality of a data set consisting of a large number of interrelated variables, while retaining as much as possible of the variation present in the principal components (PCs), which are uncorrelated, and which are ordered so that the first *few* retain most of the variation present in *all* of the original variables.", I.T. Joliffe

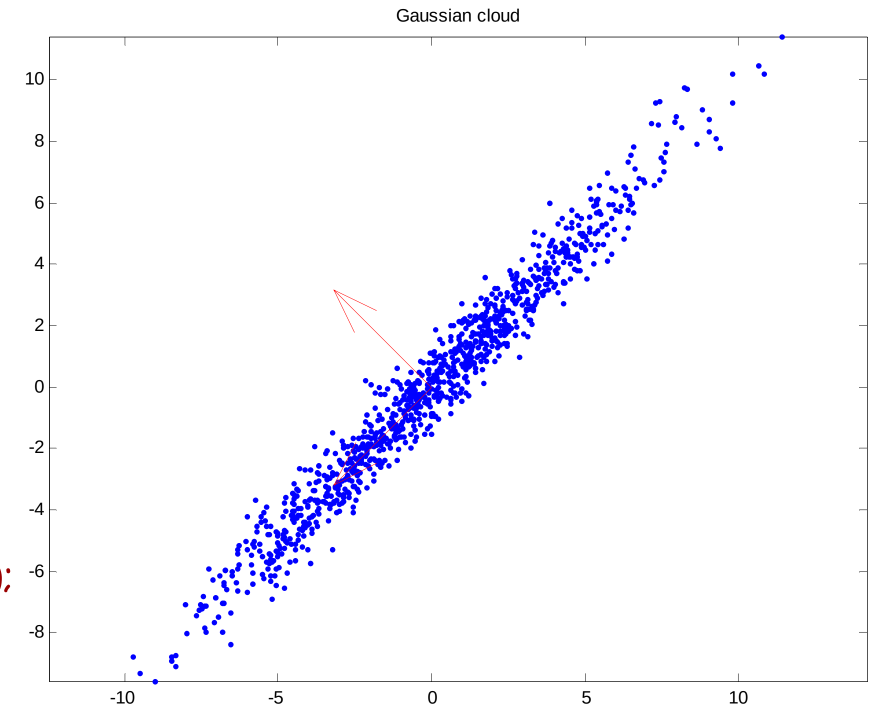
- Example applications
  - Compression
  - Noise Reduction
  - Dimensionality Reduction
    - Eigenfaces, etc.

BROWN

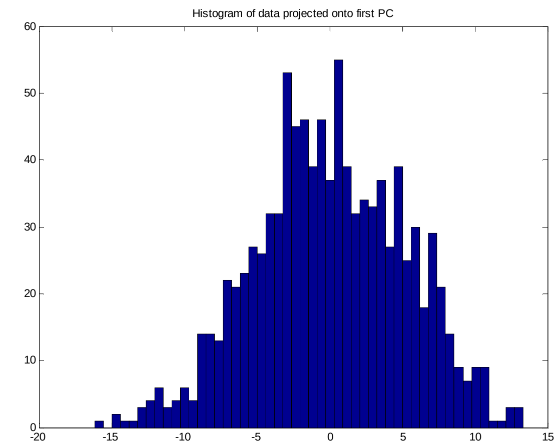
The Gist of PCA

```
num_points = 500;
angle = pi/4;
variances = [5 0; 0 .5]'
rotation = [cos(angle) -sin(angle);...
            sin(angle) cos(angle)]

data = rotation*(variances*randn(2,1000));
[pcadata,eigenvectors,eigenvalues] = pca(data,2);
recovered_rotation = eigenvectors
recovered_variances = sqrt(eigenvalues)
```




```
variances =
   5.0000        0
        0   0.5000
rotation =
   0.7071   -0.7071
   0.7071    0.7071
recovered_rotation =
  -0.7042   -0.7100
  -0.7100    0.7042
recovered_variances =
   5.1584        0
        0   0.4934
```



BROWN

Frank Wood - fwood@cs.brown.edu

The Math of PCA

- First step: Find a linear function (a projection) of a R.V. $\vec{x}$ that has maximum variance. i.e.

$$\vec{\alpha}_1^{\mathrm{T}}\vec{x} = \alpha_{11}x_1 + \alpha_{12}x_2 + \cdots + \alpha_{1p}x_p = \sum_{j=1}^{p}\alpha_{1j}x_j$$

- Second through k^{th} step: Find the subsequent uncorrelated projection with maximum variance etc. i.e. $\vec{\alpha}_2, \vec{\alpha}_3, \ldots, \vec{\alpha}_k$
- Continue until "enough variance" is accounted for or up to k, the dimensionality of $\vec{x}$.

BROWN

Finding a principal component (PC)

- Maximize the variance of the projection:

$$\arg\max_{\vec{\alpha}_1} E\left[\vec{\alpha}_1^{\mathrm{T}}\vec{x}\left(\vec{\alpha}_1^{\mathrm{T}}\vec{x}\right)^{\mathrm{T}}\right]$$

$$\arg\max_{\vec{\alpha}_1} E\left[\vec{\alpha}_1^{\mathrm{T}}\vec{x}\vec{x}^{\mathrm{T}}\vec{\alpha}_1\right]$$

$$\arg\max_{\vec{\alpha}_1} \vec{\alpha}_1^{\mathrm{T}}\Sigma\vec{\alpha}_1$$

- Easy to do! Set $\vec{\alpha}_1 = \infty$
- Solution: constrain $\vec{\alpha}_1^{\mathrm{T}}\vec{\alpha}_1 = 1$

BROWN

Constrained Optimization

- Use Lagrange multiplier and differentiate:

$$\frac{\partial}{\partial \vec{\alpha}_1}\left(\vec{\alpha}_1^{\mathrm{T}}\Sigma\vec{\alpha}_1 - \lambda\left(\vec{\alpha}_1^{\mathrm{T}}\vec{\alpha}_1 - 1\right)\right) = 0$$

$$\Sigma\vec{\alpha}_1 - \lambda\vec{\alpha}_1 = 0$$

$$(\Sigma - \lambda \mathrm{I})\vec{\alpha}_1 = 0 \quad \text{or} \quad \Sigma\vec{\alpha}_1 = \lambda\vec{\alpha}_1$$

- So $\vec{\alpha}_1$ is an eigenvector of Σ and λ is the corresponding eigenvalue.

BROWN

Optimal Properties of PC's

- The second, third, etc. PC's can be found using a similar derivation subject of course to additional constraints.
- It can be shown that a choosing B' to be the first q eigenvectors of the covariance matrix Σ of x that the orthonormal linear transformation

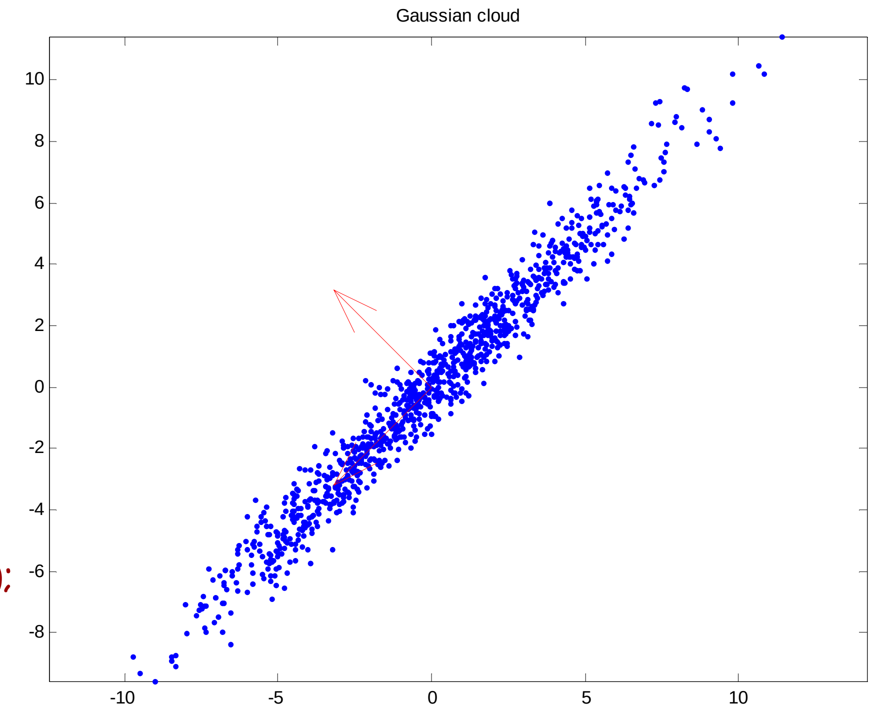
$$y = B'x$$

maximizes the covariance of y.

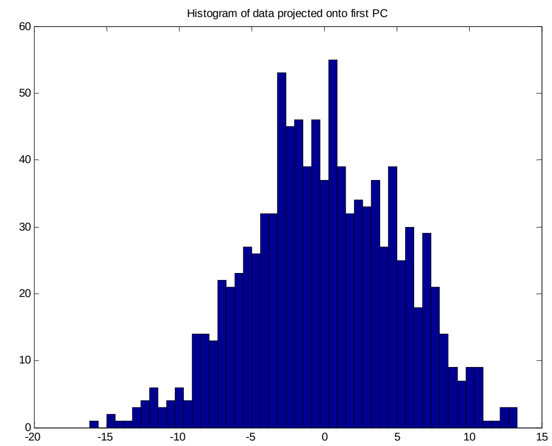
The Gist of PCA

```
num_points = 500;
angle = pi/4;
variances = [5 0; 0 .5]'
rotation = [cos(angle) -sin(angle);...
            sin(angle) cos(angle)]

data = rotation*(variances*randn(2,1000));
[pcadata,eigenvectors,eigenvalues] = pca(data,2);
recovered_rotation = eigenvectors
recovered_variances = sqrt(eigenvalues)
```




```
variances =
   5.0000        0
        0   0.5000
rotation =
   0.7071   -0.7071
   0.7071    0.7071
recovered_rotation =
  -0.7042   -0.7100
  -0.7100    0.7042
recovered_variances =
   5.1584        0
        0   0.4934
```



BROWN

Frank Wood - fwood@cs.brown.edu

EM for Gaussian Mixture Models

- Expectation Maximization is a recursive method for estimating the parameters of data distributions with missing or unobserved data.
- In our case, the "missing data" is data class memberships.

Probability of x_i assuming that it came from that class.

$$\log(L(\theta \mid \mathrm{X},\mathrm{Y})) = \log(P(\mathrm{X},\mathrm{Y} \mid \theta)) = \log \prod_{i=1}^{n} p(y_i)p\left(x_i \mid \theta_{y_i}\right)$$

Shorthand for "the prior probability of the class labeled y_i"

- This represents a generative view with latent structure.

BROWN

Frank Wood - fwood@cs.brown.edu

Closed form E & M steps for GMM

$$\alpha_l^{new} = \frac{1}{N}\sum_{i=1}^{N} p\left(l \mid x_i, \Theta^g\right)$$

$$\mu_l^{new} = \frac{\sum_{i=1}^{N} x_i\, p\left(l \mid x_i, \Theta^g\right)}{\sum_{i=1}^{N} p\left(l \mid x_i, \Theta^g\right)}$$

$$\Sigma_l^{new} = \frac{\sum_{i=1}^{N} p\left(l \mid x_i, \Theta^g\right)\left(x_i - \mu_l^{new}\right)^{\mathrm{T}}\left(x_i - \mu_l^{new}\right)}{\sum_{i=1}^{N} p\left(l \mid x_i, \Theta^g\right)}$$

From "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models", Jeff A. Bilmes

BROWN

Example Application – Spike Sorting

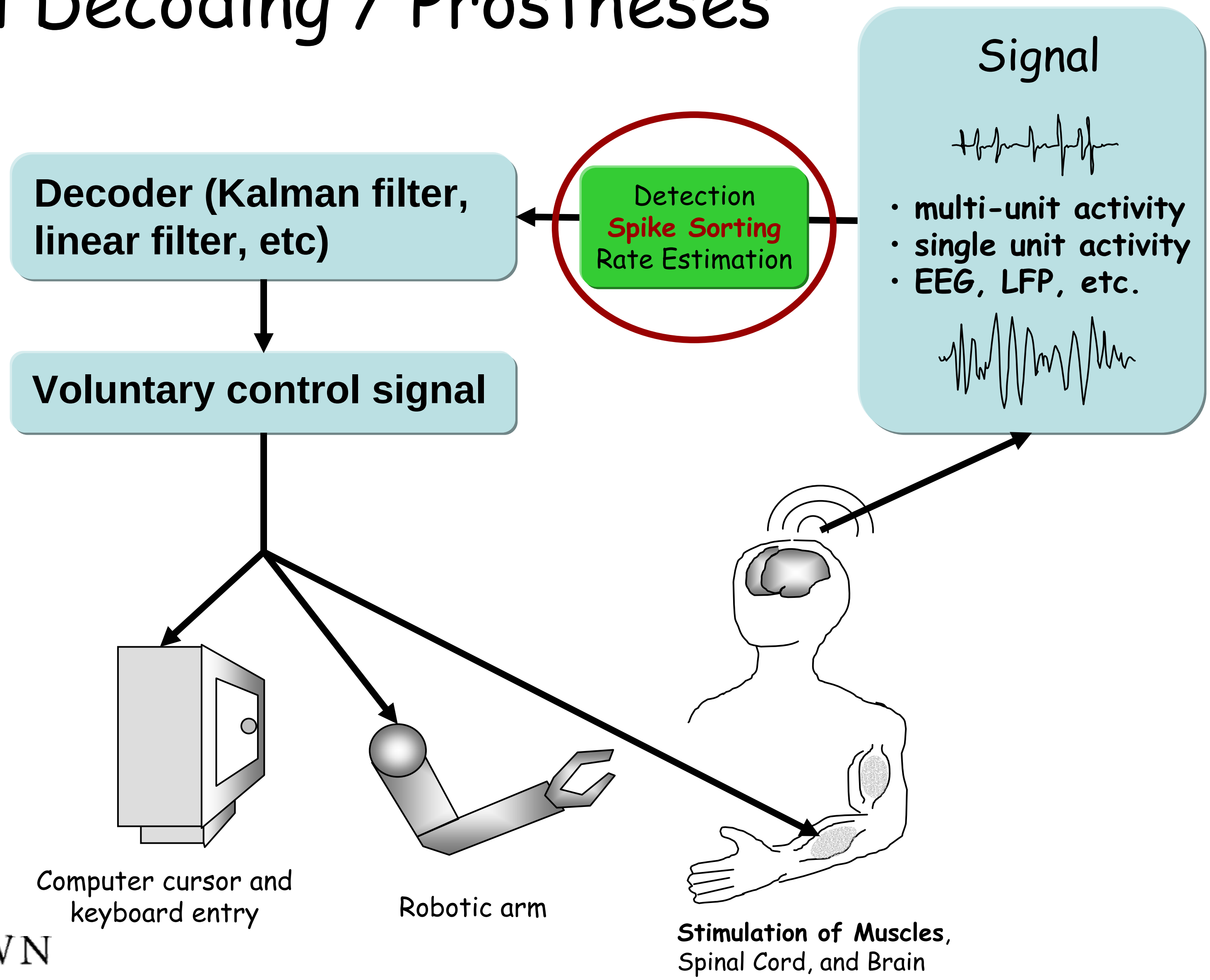
- Previous Solutions
  - Non-parametric template matching
  - Various clustering's of principle components (Lewiki)
  - EM on mixtures of multivariate t-distributions (Shoham, et al)
  - Wavelet packets (Hulata, et al)

- Problems
  - Manually determine waveform templates
  - Manually determine number of clusters
  - Manually identify noise

  - Waveform variability
  - Inter-spike interval
  - Off-line vs. on-line

BROWN

Neural Decoding / Prostheses



BROWN

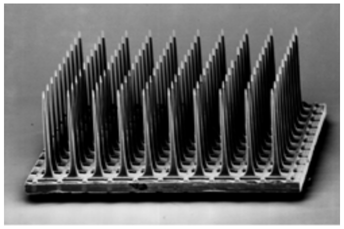
A Slight Untruth has been Implied

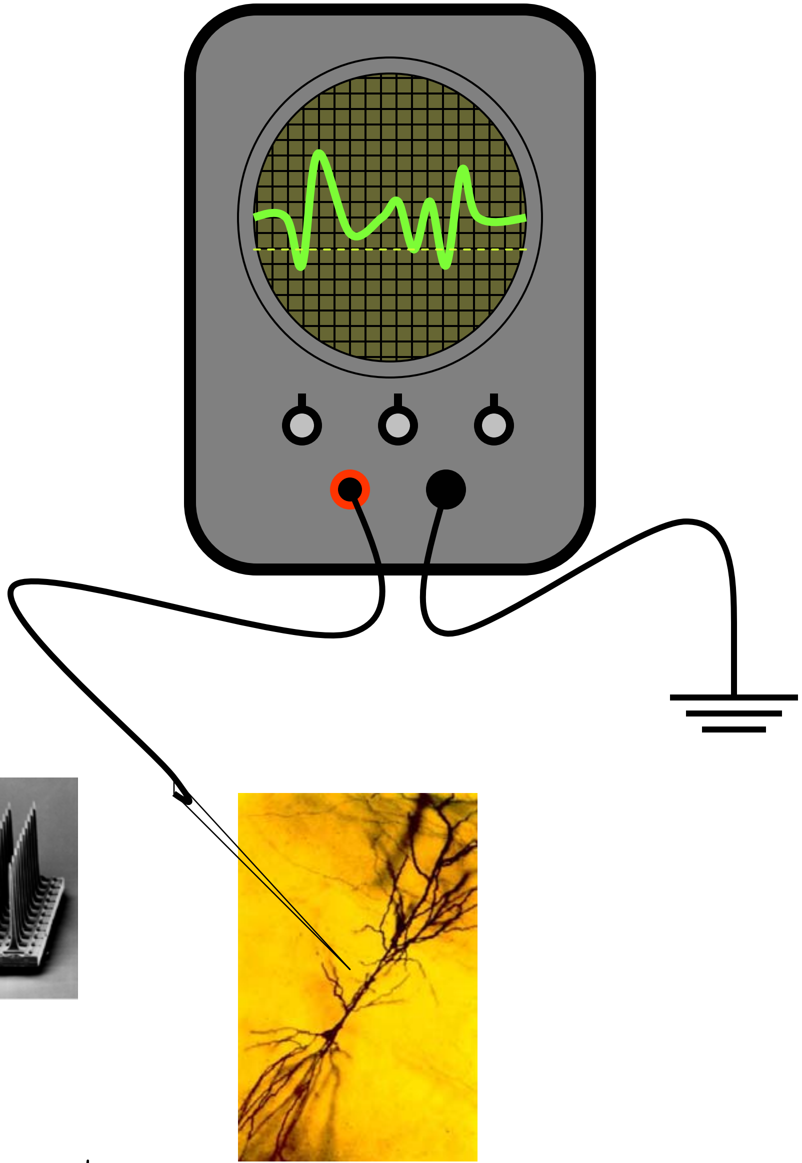
- The data you have is sorted horribly.
- Most of the 42 channels you have are "multi-units" not actually single neurons.
- It is virtually impossible to isolate and record a single neuron with perfect certainty with any recording technology and is even more difficult to do with an array due to its random insertion.

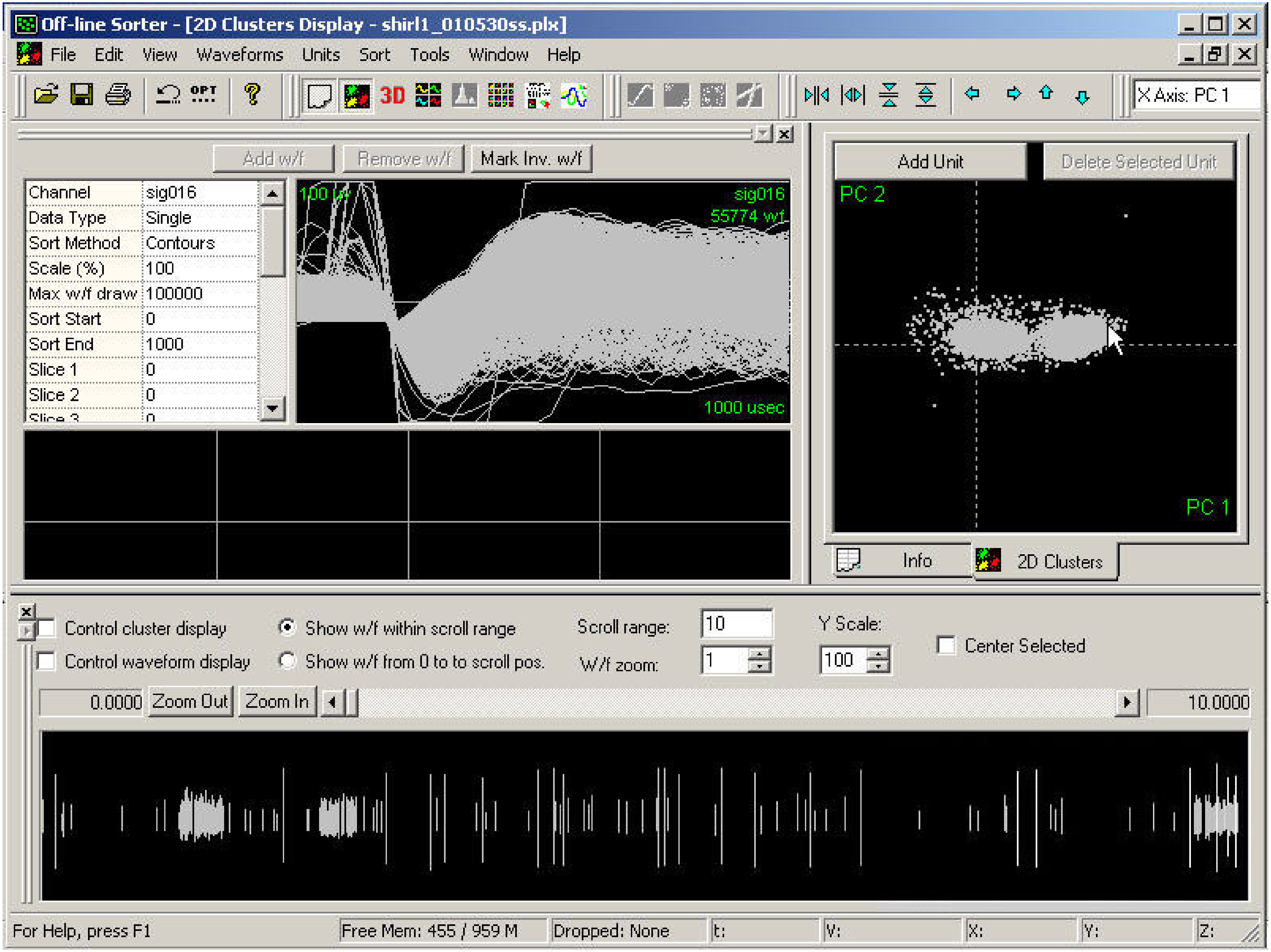
BROWN

Spike Sorting

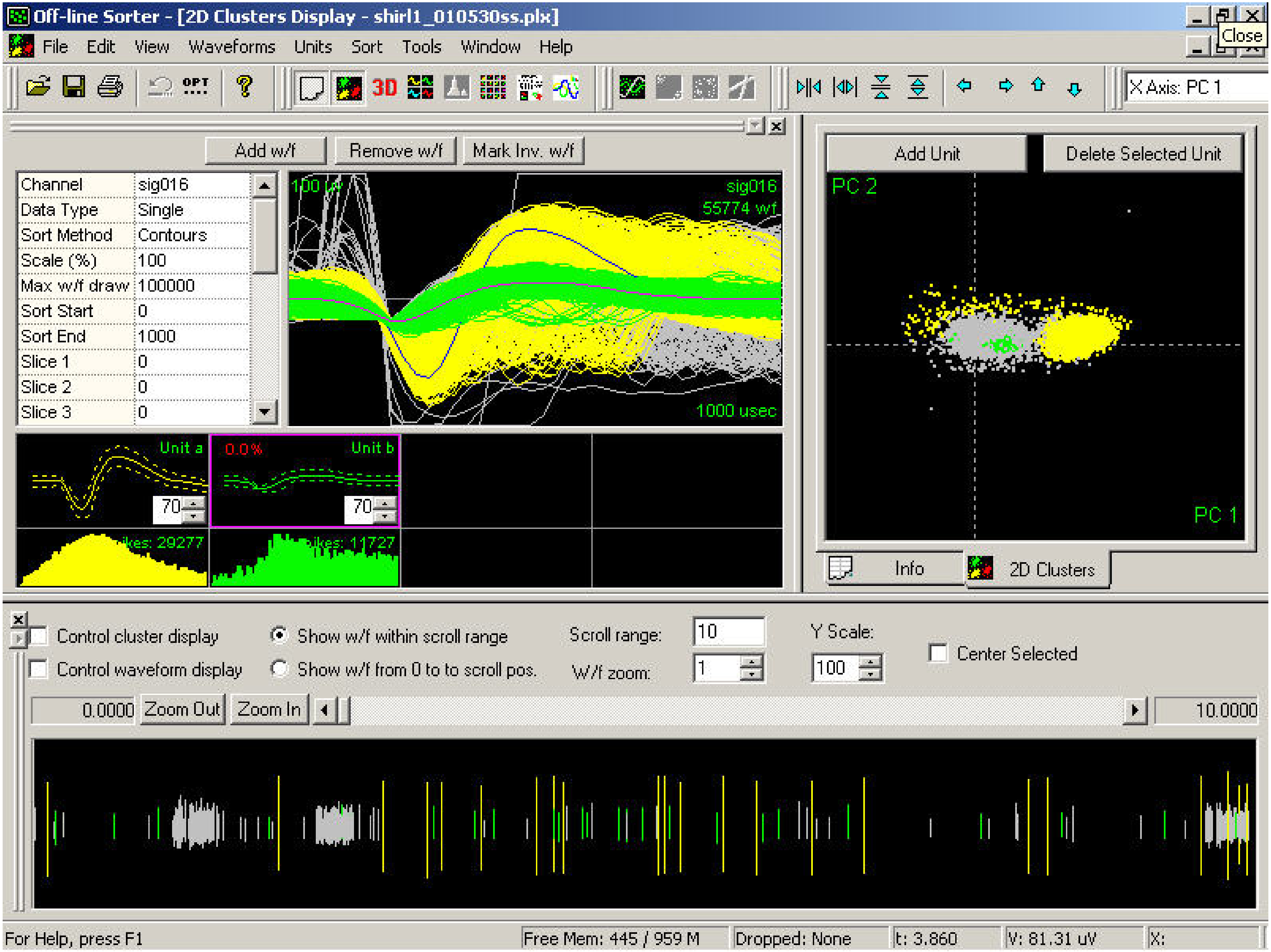
- Our definition: waveforms captured at threshold crossings are "sorted" by deciding :
  - which are "spikes"
  - how many neurons there are
  - which neurons each came from.
  - *Not detection!*
- Results from Bionic microelectrode array.





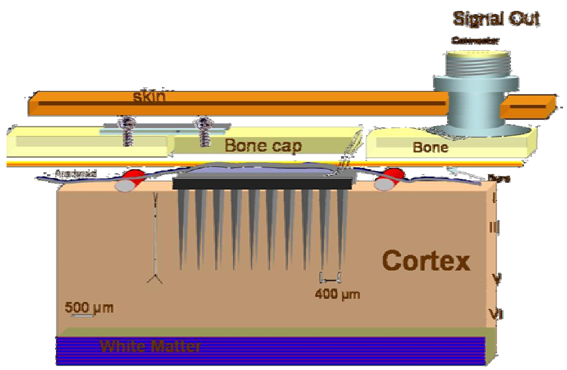


Off-line Sorter - [2D Clusters Display - shirl1_010530ss.plx]
File
Edit
View
Waveforms
Units
Sort
Tools
Window
Help
X Axis: PC 1
Add w/f
Remove w/f
Mark Inv. w/f
Channel
sig016
Data Type
Single
Sort Method
Contours
Scale (%)
100
Max w/f draw
100000
Sort Start
0
Sort End
1000
Slice 1
0
Slice 2
0
100 uV
sig016
55774 wf
1000 usec
Add Unit
Delete Selected Unit
PC 2
PC 1
Info
2D Clusters
Control cluster display
Control waveform display
Show w/f within scroll range
Show w/f from 0 to to scroll pos.
Scroll range:
10
W/f zoom:
1
Y Scale:
100
Center Selected
0.0000
Zoom Out
Zoom In
10.0000
For Help, press F1
Free Mem: 455 / 959 M
Dropped: None
t:
V:
X:
Y:
Z:


Off-line Sorter - [2D Clusters Display - shirl1_010530ss.plx]
File
Edit
View
Waveforms
Units
Sort
Tools
Window
Help
Close
3D
X Axis: PC 1
Add w/f
Remove w/f
Mark Inv. w/f
Channel
sig016
Data Type
Single
Sort Method
Contours
Scale (%)
100
Max w/f draw
100000
Sort Start
0
Sort End
1000
Slice 1
0
Slice 2
0
Slice 3
0
100 uV
sig016
55774 wf
1000 usec
Unit a
0.0%
Unit b
70
70
ikes: 29277
pikes: 11727
Add Unit
Delete Selected Unit
PC 2
PC 1
Info
2D Clusters
Control cluster display
Control waveform display
Show w/f within scroll range
Show w/f from 0 to to scroll pos.
Scroll range:
10
W/f zoom:
1
Y Scale:
100
Center Selected
0.0000
Zoom Out
Zoom In
10.0000
For Help, press F1
Free Mem: 445 / 959 M
Dropped: None
t: 3.860
V: 81.31 uV
X:

Spike Sorting's dirty little secret.

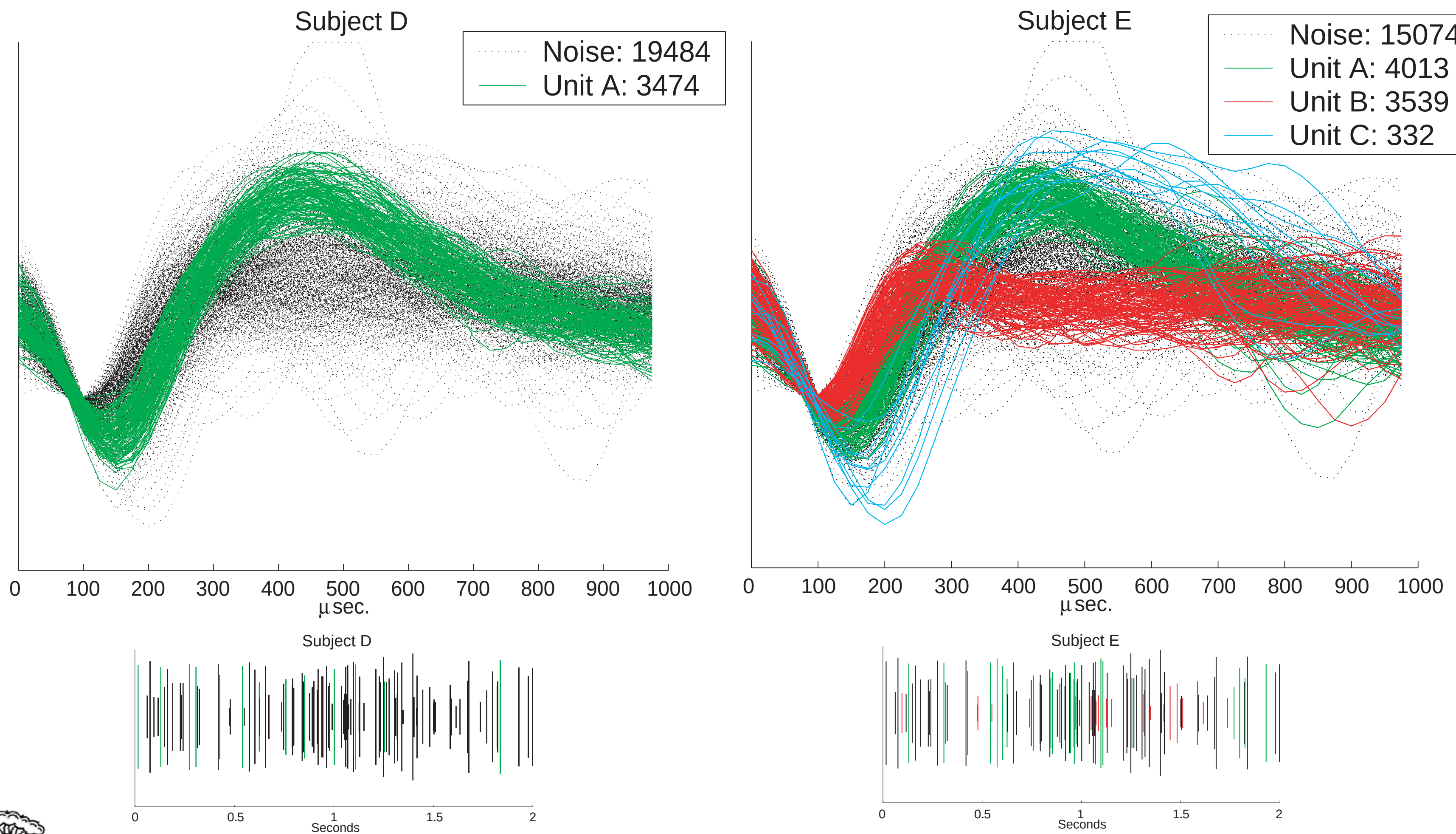
- Inspired by Harris et al (2000) we conducted a study of spike sorting subjectivity.
  - Real data
    - 5 Expert sorters
    - 20 Representative channels



Subject	A	B	C	D	E
Spikes	99160	50796	150917	77194	202351
Units	28	32	27	18	35

BROWN

Two people sorting the same channel.



BROWN

Frank Wood - fwood@cs.brown.edu

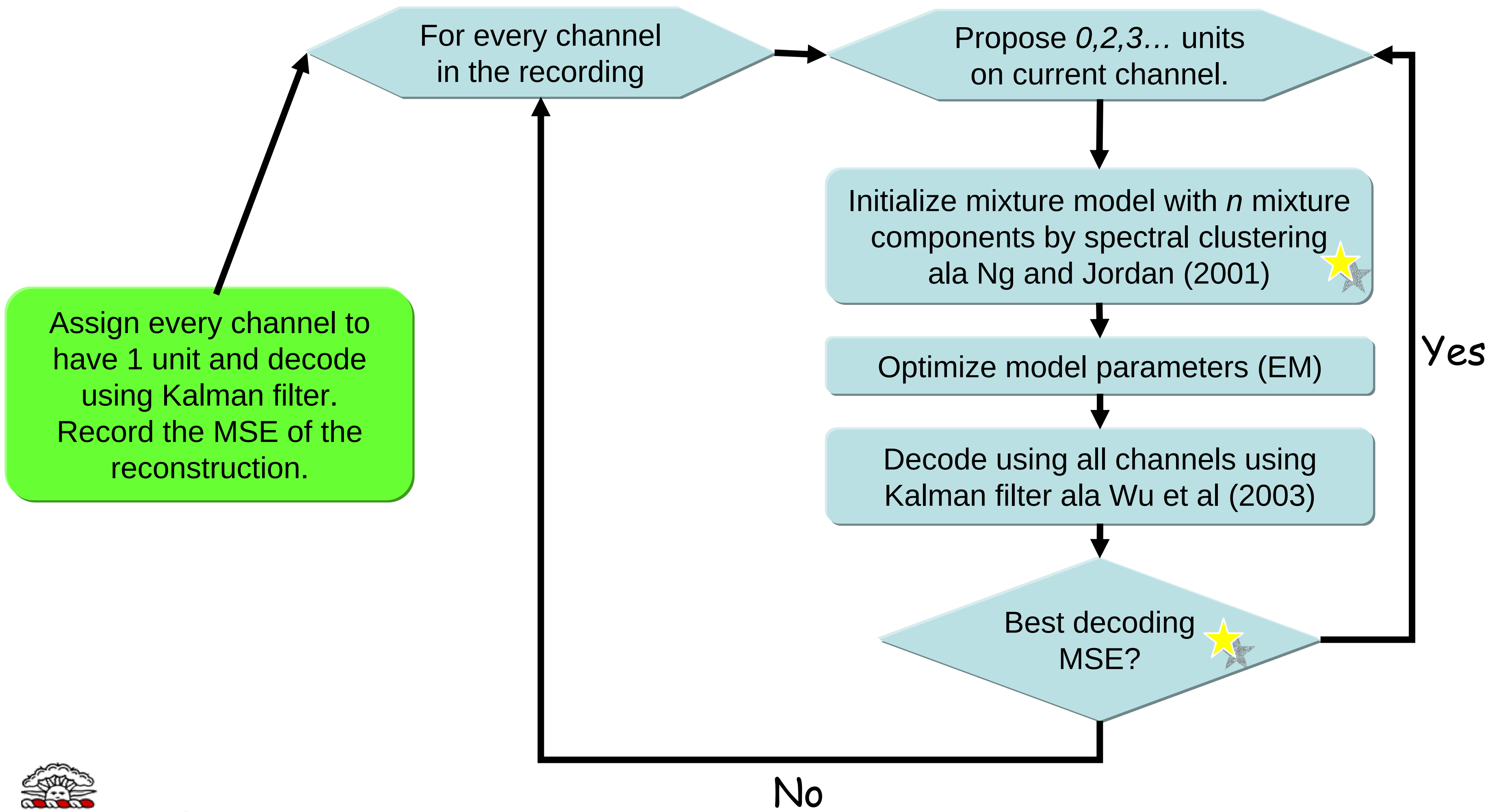
Our Goal

- Better decoding accuracy by way of improved spike sorting.

- Better spike sorting for neuroscience would be great to achieve as well but is a slightly different goal.

BROWN

A Greedy Automatic Spike Sorting Algorithm



BROWN

Frank Wood - fwood@cs.brown.edu

For full details see the paper.

Why did we talk about PCA?

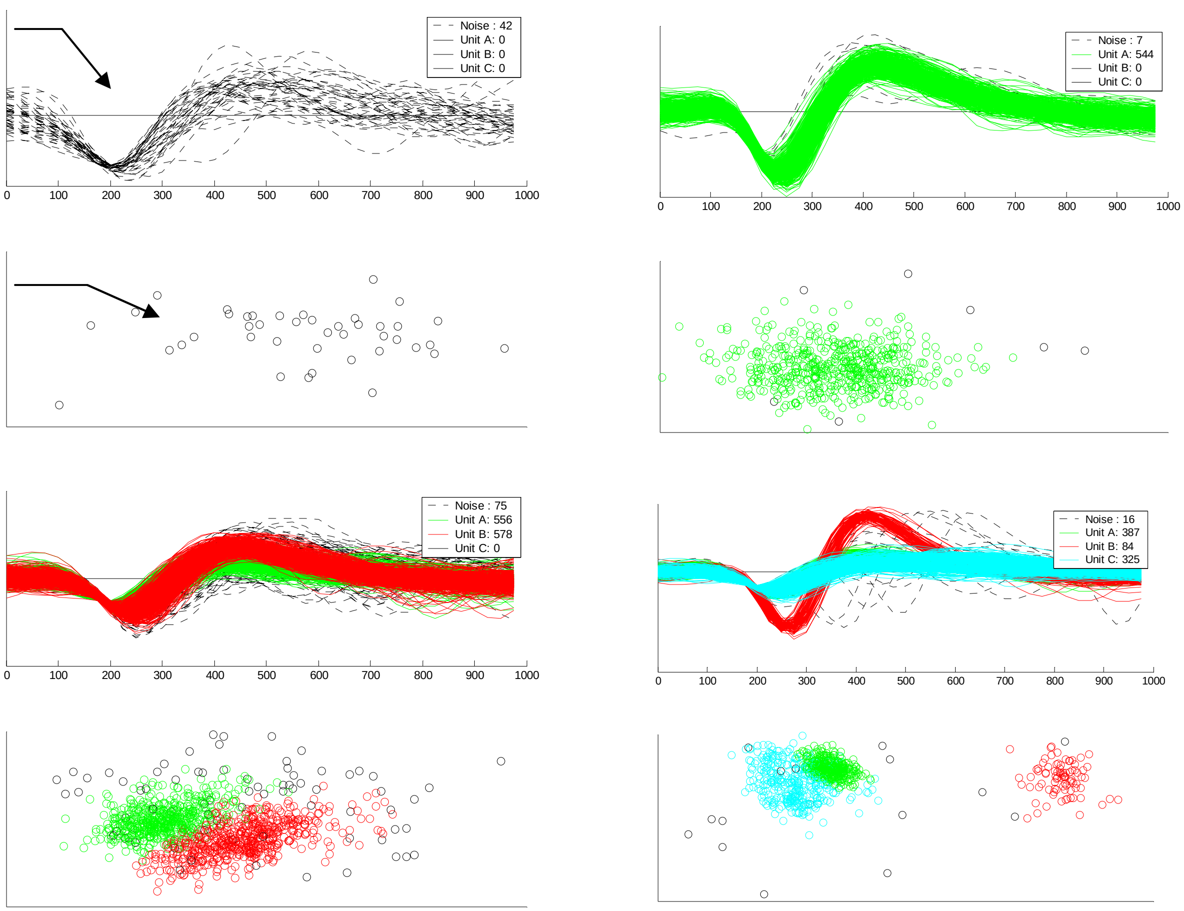
- The waveforms are largely similar (even between two different neurons).
- The intrinsic dimensionality of a waveform is probably much lower than the 48 samples we had for each.
- Speeds computation considerably and makes estimates of mixture model parameters more robust.

BROWN

Automatic Spike Sorting Visual Results

Waveforms

Corresponding 2 largest PCA coefficients.

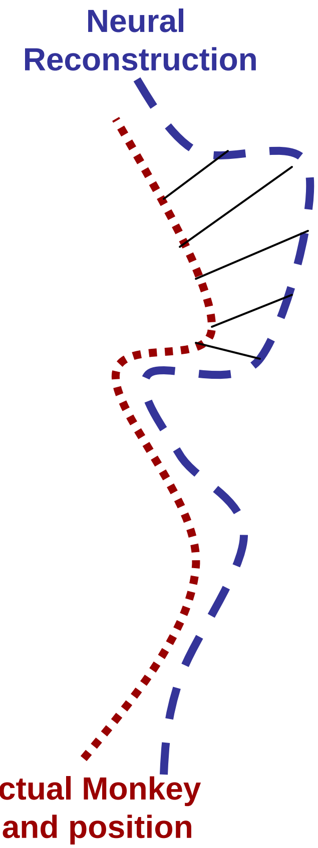


BROWN

Frank Wood - fwood@cs.brown.edu

Decoding Results

Subject	Neurons	Spikes	MSE (cm^2)
A	107	757674	11.45 +/- 1.39
B	96	335656	16.16 +/- 2.38
C	78	456221	13.37 +/- 1.52
D	88	642422	12.37 +/- 1.22
Ave. Human	92	547993	13.46 +/- 2.54



Rank: Auto Sorted → No Sorting → Randomly Sorted → Human Sorted !

BROWN

Conclusions and Discussion

- This automatic sorting algorithm produces better spike trains for neural decoding.
- Maybe spike sorting isn't necessary for good decoding?
  - Hints at using a different signal instead?
- Linking decoding to sorting may not identify physiological neurons.

- Next Steps
  - Fully leverage probabilistic interpretation for enhanced rate estimation.
  - Different cost function.
  - Extend to continuous signal.

BROWN

Next Week

- Fun!
- Crazy papers – thought provoking ethics article.

- Get the Kalman filter assignment out of the way quickly. Given what you know now it should be quite easy!

BROWN