# 10-601 Recitation

Wednesday, September 28th, 2011

Will Bishop

# Announcements

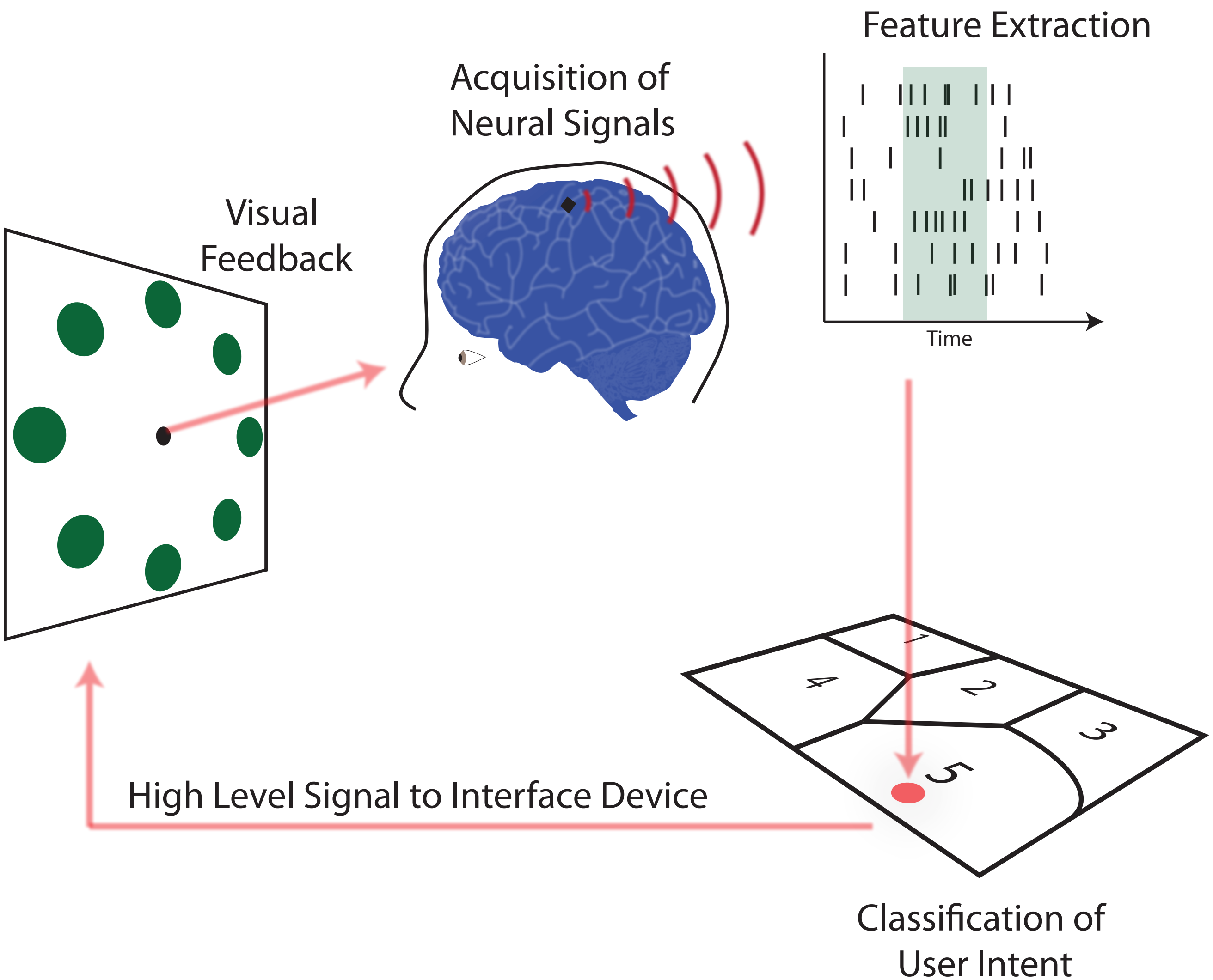
- The recitation time has been permanently set  6 - 7 PM on Wednesdays in Wean 7500 (this room)
- HW2 should be out soon.

# Topics for Today

- Conjugate Priors
- MAP Estimators - Example Derivation
- Naïve Bayes Decoders

Motivated through a real-world example from brain-computer interface (BCI).

# How would you design a decoder for this?



# Decoders we know about so far:

- Decision Trees
- Naïve Bayes

# Notation

Assume we have $U$ neurons.

$y_i$ will be label for trial $i$.

$x_{i,j}$ will be observed count for neuron $j$ for trial $i$.

# Review of Naïve Bayes

In general, we would like: $P(Y_i | X_{i,1}, \ldots X_{i,U})$.

# Review of Naïve Bayes

In general, we would like: $P(Y_i|X_{i,1}, \ldots X_{i,U})$.

Let assume we know:

1. $P(X_{i,1} \ldots X_{i,U}|Y_i)$ for $Y_i = 0$ and $Y_i = 1$.
2. $P(Y_i)$ for $Y_i = 0$ and $Y_i = 1$.

# Review of Naïve Bayes

In general, we would like: $P(Y_i|X_{i,1},\ldots X_{i,U})$.

Let assume we know:

1. $P(X_{i,1}\ldots X_{i,U}|Y_i)$ for $Y_i = 0$ and $Y_i = 1$.
2. $P(Y_i)$ for $Y_i = 0$ and $Y_i = 1$.

How do we get $P(Y_i|X_{i,1}\ldots X_{i,U})$?

Probability of target

Given observed data.

# Review of Naïve Bayes: Bayes' Rule!

Likelihood Term - We assume we know this.

$$P(Y_i|X_{i,1}\ldots X_{i,U}) = \frac{\boxed{P(X_{i,1}\ldots X_{i,U}|Y_i)}P(Y_i)}{P(X_{i,1}\ldots X_{i,U})}$$

# Review of Naïve Bayes: Bayes' Rule!

Prior Term - We assume we know this too.

$$P(Y_i|X_{i,1}\ldots X_{i,U}) = \frac{P(X_{i,1}\ldots X_{i,U}|Y_i)\boxed{P(Y_i)}}{P(X_{i,1}\ldots X_{i,U})}$$

# Review of Naïve Bayes: Bayes' Rule!

$$P(Y_i|X_{i,1}\dots X_{i,U}) = \frac{P(X_{i,1}\dots X_{i,U}|Y_i)P(Y_i)}{\boxed{P(X_{i,1}\dots X_{i,U})}}$$

Normalizing Term - Can calculate this, though in practice we often don't if all we care about is finding the class with the highest posterior probability.

# Review of Naïve Bayes: Bayes' Rule!

So if we know $P(X_{i,1} \ldots X_{i,U}|Y_i)$ and $P(Y_i)$ we can easily calculate the probabilities we need to decode with.

But how do we learn $P(X_{i,1} \ldots X_{i,U}|Y_i)$?

Let's assume that each $X_i$ value can take 10 different values. If $U = 10$, and we try to learn this using the truth table approach, how many parameters must we fit?

(10^10) - 1 Parameters!

# Review of Naïve Bayes

With naïve Bayes we assume:

$$P(X_{i,1},\ldots,X_{i,U}|Y_i) = \prod_{u=1}^{U} P(X_{i,u}|Y_i).$$

This means we can fit $U$ separate truth tables, so how many parameters do we need now?

# Review of Naïve Bayes

With naïve Bayes we assume:

$$P(X_{i,1},\ldots,X_{i,U}|Y_i) = \prod_{u=1}^{U} P(X_{i,u}|Y_i).$$

This means we can fit $U$ separate truth tables, so how many parameters do we need now?

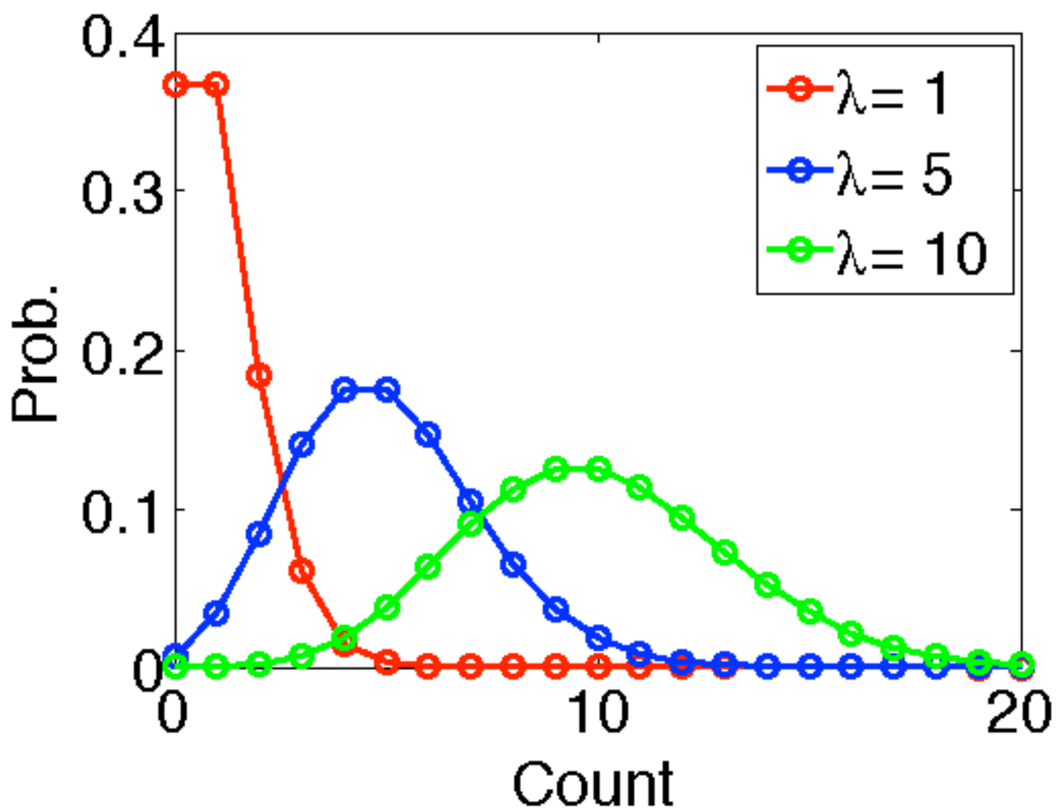
**(10-1)*10 = 90 Parameters**

Of course, we have to do this for both possible values of $Y_i$, so actually need to 180 parameters to fit $P(X_{i,1}\ldots X_{i,U}|Y_i = 0)$ and $P(X_{i,1}\ldots X_{i,U}|Y_i = 1)$.

# Motivating Example

In practice, we don't use a truth table for $P(X_i|Y_i)$ but instead assume it is a Poisson distribution.

$$P(X) = \frac{e^{-\lambda}\lambda^X}{X!}$$



# Motivating Example

So given a set of $N$ observed counts for neuron $j$ $X_{1,j} \ldots X_{N,j}$ when the subject was reaching for target $Y_i = 1$, how can we learn the appropriate $\lambda$ value for $P(X_{i,j}|Y_i = 1)$?

# Motivating Example

So given a set of $N$ observed counts for neuron $j$ $X_{1,j} \dots X_{N,j}$ when the subject was reaching for target $Y_i = 1$, how can we learn the appropriate $\lambda$ value for $P(X_{i,j}|Y_i = 1)$?

1) Maximum Likelihood Estimator (Covered last recitation)
2) Maximum A Posteriori Estimator (Covered today)

# MAP Estimators

Given a set of $N$ observations $X_1, \ldots, X_N$, we are after:

$$P(\lambda | X_1, \ldots X_N) = \frac{\underbrace{\boxed{P(X_1, \ldots, X_N | \lambda)}}_{\text{Likelihood term}} \underbrace{P(\lambda)}_{\text{Prior}}}{P(X_1, \ldots X_N)}$$

# MAP Estimators

Given a set of $N$ observations $X_1, \ldots, X_N$, we are after:

$$P(\lambda | X_1, \ldots X_N) = \frac{P(X_1, \ldots, X_N | \lambda) \boxed{P(\lambda)}}{P(X_1, \ldots X_N)}$$

Prior

# MAP Estimators

Given a set of $N$ observations $X_1, \ldots, X_N$, we are after:

$$P(\lambda | X_1, \ldots X_N) = \frac{P(X_1, \ldots, X_N | \lambda) \underbrace{\boxed{P(\lambda)}}_{\text{Prior}}}{P(X_1, \ldots X_N)}$$

How do we choose the prior?

# MAP Estimators

- Considerations when selecting the prior:
  - The prior encodes your initial beliefs (before you've seen any data) about parameter values.
  - Often, we select the prior so things work out nicely mathematically.

# Conjugate Priors

- Conjugate priors
  - A prior is conjugate to the distribution we are using for our likelihood term if:

When we multiply the the prior by the likelihood term and divide by the normalizing constant in Bayes' equation <u>the resulting probability distribution is in the same family as the prior.</u>

# Conjugate Priors

- Conjugate priors
  - A prior is conjugate to the distribution we are using for our likelihood term if:

When we multiply the the prior by the likelihood term and divide by the normalizing constant in Bayes' equation <u>the resulting probability distribution is in the same family as the prior.</u>

It makes the math easy :)

# MAP Estimator: An Example

Assume we have $X_1, \ldots X_N$ observations from a Poisson distribution. With unknown $\lambda$.

Let's find a MAP estimator for lambda.

# MAP Estimator: An Example

Assume we have $X_1, \ldots X_N$ observations from a Poisson distribution. With unknown $\lambda$.

Assume our prior belief on $\lambda$ is given by a Gamma distribution.

In other words: $\lambda \sim \mathrm{Gamma}(\alpha, \beta)$

# MAP Estimator: An Example

$P(\lambda) \sim \text{Gamma}(\alpha, \beta)$

The pdf for a $\text{Gamma}(\alpha, \beta)$ distribution is:

$$P(\lambda) = \frac{1}{\Gamma(\alpha)\beta^\alpha} \lambda^{\alpha-1} e^{-\lambda/\beta}$$

# MAP Estimator: An Example

$P(\lambda) \sim \mathrm{Gamma}(\alpha, \beta)$

The pdf for a $\mathrm{Gamma}(\alpha, \beta)$ distribution is:

$$P(\lambda) = \frac{1}{\Gamma(\alpha)\beta^{\alpha}} \lambda^{\alpha-1} e^{-\lambda/\beta}$$

Just a normalizing constant.

# MAP Estimator: An Example

$P(\lambda) \sim \mathrm{Gamma}(\alpha, \beta)$

The pdf for a $\mathrm{Gamma}(\alpha, \beta)$ distribution is:

$$\begin{aligned} P(\lambda) &= \frac{1}{\Gamma(\alpha)\beta^{\alpha}} \lambda^{\alpha-1} e^{-\lambda/\beta} \\ &= \frac{1}{C} \lambda^{\alpha-1} e^{-\lambda/\beta} \end{aligned}$$

# MAP Estimator: An Example

Let's write out the likelihood for our data.

## Let's write out the likelihood for our data.

$$P(x_1, \ldots, x_N | \lambda) =$$

Let's write out the likelihood for our data.

$$P(x_1, \ldots, x_N | \lambda) = \prod_{n=1}^{N} P(x_n | \lambda)$$

# Let's write out the likelihood for our data.

$$
\begin{aligned}
P(x_1, \ldots, x_N | \lambda) &= \prod_{n=1}^{N} P(x_n | \lambda) \\
&= \prod_{n=1}^{N} \frac{e^{-\lambda} \lambda^{x_n}}{x_n!}
\end{aligned}
$$

Let's write out the likelihood for our data.

$$
\begin{aligned}
P(x_1, \ldots, x_N | \lambda) &= \prod_{n=1}^{N} P(x_n | \lambda) \\
&= \prod_{n=1}^{N} \frac{e^{-\lambda} \lambda^{x_n}}{x_n!} \\
&= \frac{\left( \prod_{n=1}^{N} e^{-\lambda} \right) \left( \prod_{n=1}^{N} \lambda^{x_n} \right)}{\prod_{n=1}^{N} x_n!}
\end{aligned}
$$

Let's write out the likelihood for our data.

$$
\begin{aligned}
P(x_1, \ldots, x_N|\lambda) &= \prod_{n=1}^{N} P(x_n|\lambda) \\
&= \prod_{n=1}^{N} \frac{e^{-\lambda}\lambda^{x_n}}{x_n!} \\
&= \frac{\left(\prod_{n=1}^{N} e^{-\lambda}\right)\left(\prod_{n=1}^{N} \lambda^{x_n}\right)}{\prod_{n=1}^{N} x_n!} \\
&= \frac{\left(\ e^{-\lambda}\right)^{N} \lambda^{(x_1+x_2+\ldots+x_n)}}{\prod_{n=1}^{N} x_n!}
\end{aligned}
$$

## Let's write out the likelihood for our data.

$$
\begin{aligned}
P(x_1, \ldots, x_N | \lambda) &= \prod_{n=1}^{N} P(x_n | \lambda) \\
&= \prod_{n=1}^{N} \frac{e^{-\lambda} \lambda^{x_n}}{x_n!} \\
&= \frac{\left( \prod_{n=1}^{N} e^{-\lambda} \right) \left( \prod_{n=1}^{N} \lambda^{x_n} \right)}{\prod_{n=1}^{N} x_n!} \\
&= \frac{\left( e^{-\lambda} \right)^{N} \lambda^{(x_1 + x_2 + \ldots + x_n)}}{\prod_{n=1}^{N} x_n!} \\
&= \frac{e^{-N\lambda} \lambda^{\left( \sum_{n=1}^{N} x_n \right)}}{\prod_{n=1}^{N} x_n!}
\end{aligned}
$$

Let's write out the likelihood for our data.

$$P(x_1, \ldots, x_N | \lambda) = \prod_{n=1}^{N} P(x_n | \lambda)$$

$$= \prod_{n=1}^{N} \frac{e^{-\lambda} \lambda^{x_n}}{x_n!}$$

$$= \frac{\left( \prod_{n=1}^{N} e^{-\lambda} \right) \left( \prod_{n=1}^{N} \lambda^{x_n} \right)}{\prod_{n=1}^{N} x_n!}$$

$$= \frac{\left( e^{-\lambda} \right)^N \lambda^{(x_1 + x_2 + \ldots + x_n)}}{\prod_{n=1}^{N} x_n!}$$

$$\boxed{= \frac{e^{-N\lambda} \lambda^{\left( \sum_{n=1}^{N} x_n \right)}}{\prod_{n=1}^{N} x_n!}}$$

Our final likelihood term.

# MAP Estimator: An Example

Put everything into Baye's equation:

$$P(\lambda|X_1,\dots X_N) = \frac{P(X_1,\dots,X_N|\lambda)P(\lambda)}{P(X_1,\dots X_N)}$$

$$P(\lambda|x_1,\dots,x_n) = \frac{\quad}{\quad}$$

# MAP Estimator: An Example

Put everything into Baye's equation:

$$P(\lambda|X_1, \ldots X_N) = \frac{P(X_1, \ldots, X_N|\lambda)P(\lambda)}{P(X_1, \ldots X_N)}$$

$$P(\lambda|x_1, \ldots, x_n) = \frac{\left(\frac{e^{-N\lambda}\lambda^{\left(\sum_{n=1}^{N} x_n\right)}}{\prod_{n=1}^{N} x_n!}\right)}{\phantom{X}}$$

# MAP Estimator: An Example

Put everything into Baye's equation:

$$P(\lambda|X_1,\ldots X_N) = \frac{P(X_1,\ldots,X_N|\lambda)P(\lambda)}{P(X_1,\ldots X_N)}$$

$$P(\lambda|x_1,\ldots,x_n) = \frac{\left(\frac{e^{-N\lambda}\lambda^{\left(\sum_{n=1}^{N} x_n\right)}}{\prod_{n=1}^{N} x_n!}\right)\frac{1}{\Gamma(\alpha)\beta^\alpha}\lambda^{\alpha-1}e^{-\lambda/\beta}}{}$$

# MAP Estimator: An Example

Put everything into Baye's equation:

$$P(\lambda|X_1, \dots X_N) = \frac{P(X_1, \dots, X_N|\lambda)P(\lambda)}{P(X_1, \dots X_N)}$$

$$P(\lambda|x_1, \dots, x_n) = \frac{\left(\frac{e^{-N\lambda}\lambda^{\left(\sum_{n=1}^{N} x_n\right)}}{\prod_{n=1}^{N} x_n!}\right)\frac{1}{\Gamma(\alpha)\beta^{\alpha}}\lambda^{\alpha-1}e^{-\lambda/\beta}}{??}$$

# MAP Estimator: An Example

$$P(x_1, \ldots, x_N) = \int_0^\infty \left( \frac{e^{-N\lambda} \lambda^{\left(\sum_{n=1}^N x_n\right)}}{\prod_{n=1}^N x_n!} \right) \frac{1}{\Gamma(\alpha)\beta^\alpha} \lambda^{\alpha-1} e^{-\lambda/\beta} d\lambda$$

# MAP Estimator: An Example

$$P(x_1,\ldots,x_N)=\int_0^\infty\left(\frac{e^{-N\lambda}\lambda^{\left(\sum_{n=1}^N x_n\right)}}{\prod_{n=1}^N x_n!}\right)\frac{1}{\Gamma(\alpha)\beta^\alpha}\lambda^{\alpha-1}e^{-\lambda/\beta}d\lambda$$

**WHAT!! I thought this was suppose to make the math nice?**

# MAP Estimator: An Example

$$P(x_1, \ldots, x_N) = \int_0^\infty \left( \frac{e^{-N\lambda} \lambda^{\left(\sum_{n=1}^N x_n\right)}}{\prod_{n=1}^N x_n!} \right) \frac{1}{\Gamma(\alpha)\beta^\alpha} \lambda^{\alpha-1} e^{-\lambda/\beta} d\lambda$$

**WHAT!! I thought this was suppose to make the math nice?**

**Don't worry - we don't actually have to compute this :)**

# MAP Estimator: An Example

$$P(\lambda|x_1,\ldots,x_n) = \frac{\left(\frac{e^{-N\lambda}\lambda^{\left(\sum_{n=1}^{N} x_n\right)}}{\prod_{n=1}^{N} x_n!}\right)\frac{1}{\Gamma(\alpha)\beta^{\alpha}}\lambda^{\alpha-1}e^{-\lambda/\beta}}{??}$$

This is just a normalizing constant

# MAP Estimator: An Example

$$P(\lambda|x_1,\ldots,x_n) = \frac{\left(\frac{e^{-N\lambda}\lambda^{\left(\sum_{n=1}^{N} x_n\right)}}{\prod_{n=1}^{N} x_n!}\right)\frac{1}{\Gamma(\alpha)\beta^{\alpha}}\lambda^{\alpha-1}e^{-\lambda/\beta}}{C}$$

This is just a normalizing constant

# MAP Estimator: An Example

$$P(\lambda|x_1,\ldots,x_n) = \frac{e^{-N\lambda}\lambda^{\left(\sum_{n=1}^{N} x_n\right)}\lambda^{\alpha-1}e^{-\lambda/\beta}}{C\left(\prod_{n=1}^{N} x_n!\right)\Gamma(\alpha)\beta^{\alpha}}$$

$$= \frac{e^{-N\lambda}\lambda^{\left(\sum_{n=1}^{N} x_n\right)}\lambda^{\alpha-1}e^{-\lambda/\beta}}{D}$$

In fact, let's group every term that does not depend on lambda with the normalizing constant.

# MAP Estimator: An Example

Now, let's group terms together:

$$P(\lambda|x_1, \ldots, x_n) = \frac{1}{D} e^{-N\lambda} \lambda^{\left(\sum_{n=1}^{N} x_n\right)} \lambda^{\alpha-1} e^{-\lambda/\beta}$$

# MAP Estimator: An Example

Now, let's group terms together:

$$P(\lambda|x_1,\ldots,x_n) = \frac{1}{D}e^{-N\lambda}\lambda^{\left(\sum_{n=1}^N x_n\right)}\lambda^{\alpha-1}e^{-\lambda/\beta}$$
$$= \frac{1}{D}e^{-N\lambda-\lambda/\beta}\lambda^{\left(\sum_{n=1}^N x_n\right)+\alpha-1}$$

# MAP Estimator: An Example

Now, let's group terms together:

$$
\begin{aligned}
P(\lambda|x_1,\ldots,x_n) &= \frac{1}{D} e^{-N\lambda} \lambda^{\left(\sum_{n=1}^{N} x_n\right)} \lambda^{\alpha-1} e^{-\lambda/\beta} \\
&= \frac{1}{D} e^{-N\lambda-\lambda/\beta} \lambda^{\left(\sum_{n=1}^{N} x_n\right)+\alpha-1} \\
&= \frac{1}{D} e^{-\lambda(N+1/\beta)} \lambda^{\left(\sum_{n=1}^{N} x_n\right)+\alpha-1}
\end{aligned}
$$

# MAP Estimator: An Example

Now, let's group terms together:

$$
\begin{aligned}
P(\lambda|x_1,\ldots,x_n) &= \frac{1}{D} e^{-N\lambda} \lambda^{\left(\sum_{n=1}^{N} x_n\right)} \lambda^{\alpha-1} e^{-\lambda/\beta} \\
&= \frac{1}{D} e^{-N\lambda-\lambda/\beta} \lambda^{\left(\sum_{n=1}^{N} x_n\right)+\alpha-1} \\
&= \frac{1}{D} e^{-\lambda(N+1/\beta)} \lambda^{\left(\sum_{n=1}^{N} x_n\right)+\alpha-1} \\
&= \frac{1}{D} e^{-\lambda/\left(\frac{\beta}{N\beta+1}\right)} \lambda^{\left(\sum_{n=1}^{N} x_n\right)+\alpha-1}
\end{aligned}
$$

# MAP Estimator: An Example

$$P(\lambda|x_1,\ldots,x_n) = \frac{1}{D}e^{-\lambda/(\frac{\beta}{N\beta+1})}\lambda^{\left(\sum_{n=1}^{N} x_n\right)+\alpha-1}$$

Compare this to the form of a Gamma distribution.

# MAP Estimator: An Example

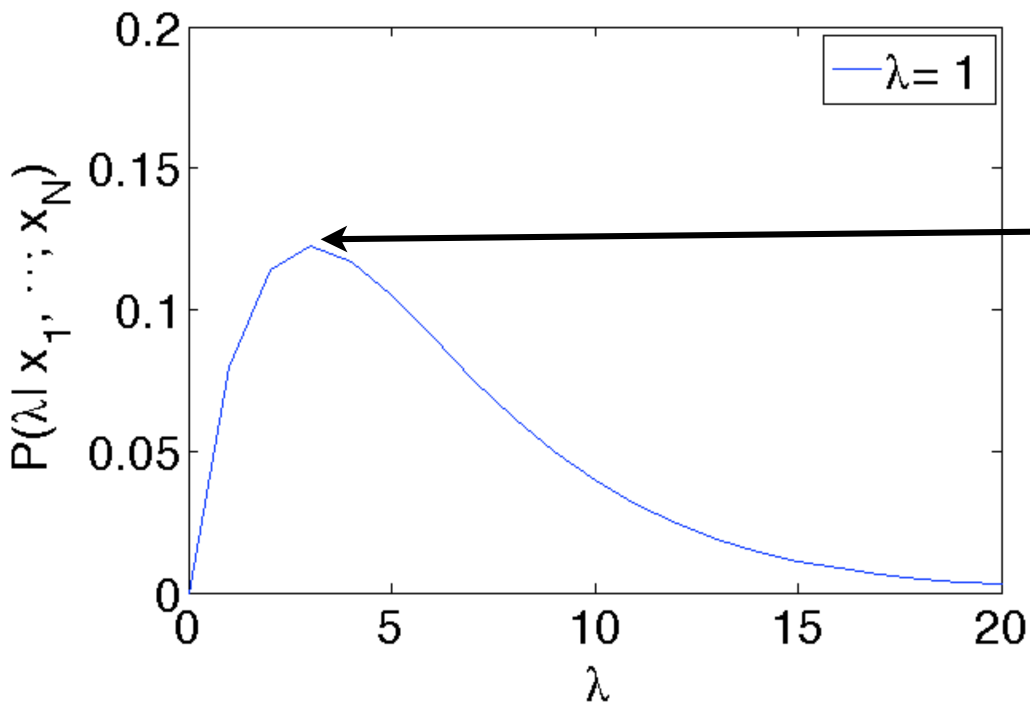
$$P(\lambda|x_1, \ldots, x_n) = \frac{1}{D} e^{-\lambda/(\frac{\beta}{N\beta+1})} \lambda^{\left(\sum_{n=1}^{N} x_n\right)+\alpha-1}$$

Compare this to the form of a Gamma distribution.

We have a Gamma $\left(\alpha + \sum_{n=1}^{N} x_n, \frac{\beta}{N\beta+1}\right)$ for the posterior.

# MAP Estimator: An Example

$$\lambda | x_1, \ldots x_n \sim \text{Gamma}\left(\alpha + \sum_{n=1}^{N} x_n, \frac{\beta}{N\beta + 1}\right)$$




Fact: the mode of a Gamma$(\alpha, \beta)$ distribution is located at $(\alpha - 1)\beta$.

# MAP Estimator: An Example

$$\lambda | x_1, \ldots x_n \sim \text{Gamma}\left(\alpha + \sum_{n=1}^{N} x_n, \frac{\beta}{N\beta + 1}\right)$$

Fact: the mode of a Gamma$(\alpha, \beta)$ distribution is located at $(\alpha - 1)\beta$.

So the mode of our distribution is located at: $(\alpha + \sum_{n=1}^{N} x_n - 1)\left(\frac{\beta}{N\beta+1}\right)$.

# MAP Estimator: An Example

$$\lambda | x_1, \ldots x_n \sim \text{Gamma}\left(\alpha + \sum_{n=1}^{N} x_n, \frac{\beta}{N\beta + 1}\right)$$

Fact: the mode of a Gamma$(\alpha, \beta)$ distribution is located at $(\alpha - 1)\beta$.

Thus, our map estimator for $\lambda$ is:

$$\hat{\lambda} = (\alpha + \sum_{n=1}^{N} x_n - 1)\left(\frac{\beta}{N\beta + 1}\right)$$

# Brief Aside: Naïve Bayes' (and a whole lot of work….) might get you a *Nature* Paper!

Vol 442|13 July 2006|**doi:10.1038/nature04968** nature

LETTERS

## A high-performance brain–computer interface

Gopal Santhanam[1]*, Stephen I. Ryu[1,2]*, Byron M. Yu[1], Afsheen Afshar[1,3] & Krishna V. Shenoy[1,4]

**Recent studies have demonstrated that monkeys[1–4] and humans[5–9] can use signals from the brain to guide computer cursors. Brain–computer interfaces (BCIs) may one day assist patients suffering from neurological injury or disease, but relatively low system performance remains a major obstacle. In fact, the speed and accuracy with which keys can be selected using BCIs is still far lower than for systems relying on eye movements. This is true whether BCIs use recordings from populations of individual neurons using invasive electrode techniques[1–5,7,8] or electroencephalogram recordings using less-[6] or non-invasive[9] techniques. Here we present the design and demonstration, using electrode arrays implanted in monkey dorsal premotor cortex, of a manyfold higher performance BCI than previously reported[9,10]. These results indicate that a fast and accurate key selection system, capable of operating with a range of keyboard sizes, is possible (up to 6.5 bits per second, or ~15 words per minute, with 96 electrodes). The highest information throughput is achieved with unprecedentedly brief neural recordings, even as recording quality degrades over time. These performance results and their implications for system design should substantially increase the clinical viability of BCIs in humans.**

Most BCIs translate neural activity into a continuous movement command, which guides a computer cursor to a desired visual target[1–3,5–9]. If the cursor is used to select targets representing discrete actions, the BCI serves as a communication prosthesis. Examples include typing keys on a keyboard, turning on room lights, and moving a wheelchair in specific directions. Human-operated BCIs are currently capable of communicating only a few letters per minute (~1 bits per second (bps) sustained rate[9]) and monkey-operated systems can only accurately select one target every 1–3 s (~1.6 bps sustained rate[10]), despite using invasive electrodes.

An alternative, potentially higher-performance approach is to translate neural activity into a prediction of the intended target and immediately place the cursor directly on that location. This type of control is appropriate for communication prostheses and benefits from not having to estimate unnecessary parameters such as continuous trajectory[4,11]. We conducted a series of experiments to investigate how quickly and accurately a BCI could operate under direct end-point control.

We used a standard instructed-delay behavioural task[12] to assess neural activity in the arm representation of monkey premotor cortex (PMd), as shown in Fig. 1a and described in Methods. As previously

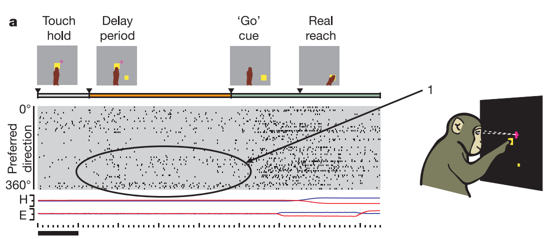



**Figure 1 | Instructed-delay (real reach) and BCI (prosthetic cursor) tasks, with accompanying neural data.** Large numbered ellipses draw attention to the increase in neural activity related to the peripheral reach target. **a,** Standard instructed-delay reach trial. Data from selected neural units are shown (grey shaded region); each row corresponds to one unit and black tick marks indicate spike times. Units are ordered by angular tuning direction (preferred direction) during the delay period. For hand (H) and eye (E) traces, blue and red lines show the horizontal and vertical coordinates, respectively. The full range of scale for these data is ±15 cm from the centre touch cue. **b,** Chain of three prosthetic cursor trials followed by a

Santhanam G*, Ryu SI*, Yu BM, Afshar A, Shenoy KV (2006) A high-performance brain-computer interface. Nature. 442:195-198.

*To be totally fair: Only one of two monkeys used Naïve Bayes decoder, but still…. :)