Lab #3 - Bayesian estimation of motion

NPB261B - Winter 2005
Instructor: Bruno A. Olshausen
Due: Monday, March 21, 2005

In this assignment we will use Bayes’ rule to infer the velocity of a complex stimulus (motion plaid) from local spatial and temporal derivatives of the image pixels, following the method described in Weiss et al. (2002). First, we will examine a very simple case of Bayesian inference with Gaussian variables, which helps to illustrate the basic principles at work in the Weiss et al. paper. Then we will run a simulation of the Weiss et al. model applied to a motion plaid.

1 Simple example

Let’s say that we say we are attempting to estimate some underlying variable, $x$, which is corrupted by additive Gaussian noise, $n$, leading to observation $y$:

$$y = x + n$$

We wish to recover $x$ given the measurement, $y$. Obviously if you don’t know the value of $n$ then there is not a unique solution - you have to guess. But as long as you know something about what values of $x$ and $n$ are typical—i.e., the probability distributions $P(x)$ and $P(n)$—then we can make an educated guess about the value of $x$. That is, we can infer its value by combining the data, $y$, with prior knowledge.

According to Bayesian inference, we first need to calculate the posterior, $P(x|y)$, and then from this we estimate the value $x$, for example by choosing the value with maximum probability. We can calculate $P(x|y)$ using Bayes’ rule as follows:

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}.$$

The two main things we need to specify here are $P(y|x)$ and $P(x)$. $P(y)$ plays the role of a normalizing constant in this case and can be ignored.

Let’s first consider $P(y|x)$—i.e., the probability of $y$ given $x$. The only uncertainty here is due to the noise, $n$. If the noise is Gaussian with zero mean then its distribution is

$$P(n) \propto e^{-\frac{n^2}{2\sigma^2}}.$$
where $\sigma_n$ is the standard deviation of the noise. Thus for $P(y|x)$ we have

$$P(y|x) = P(n + x|x)$$

$$\propto e^{-\frac{(y-x)^2}{2\sigma_n^2}}.$$  \hfill (4)

For the prior, $P(x)$, let’s also assume a Gaussian with zero mean, but with standard deviation $\sigma_x$:

$$P(x) \propto e^{-\frac{x^2}{2\sigma_x^2}}.$$  \hfill (5)

Thus, for the posterior we have

$$P(x|y) \propto P(y|x)P(x)$$

$$\propto e^{-\frac{(y-x)^2}{2\sigma_n^2}} e^{-\frac{x^2}{2\sigma_x^2}}.$$  \hfill (6)

If we wish to find the maximum-a-posteriori estimate of $x$ (the value with maximum probability), then we need to solve for the $x$ that maximizes this expression. We can make life easier by taking the logarithm of $P(x|y)$, which gets rid of the exponentials:

$$\log P(x|y) = -\frac{(y-x)^2}{2\sigma_n^2} - \frac{x^2}{2\sigma_x^2} + \text{const.}$$  \hfill (7)

Since log is a monotonically increasing function, the value of $x$ that maximizes this expression is the same as in the one above (eq. 8). You can solve for $x$ in eq. 9 by differentiating with respect to $x$ and setting the result to zero. If you re-arrange terms so that $x$ is on the left-hand side, this should yield an expression in which $x$ is computed as a function of $\sigma_x$, $\sigma_n$, and $y$ as follows:

$$\hat{x} = g(\sigma_x, \sigma_n) y$$  \hfill (8)

In other words, the estimate for $x$, which we denote $\hat{x}$, is a linear function of $y$, with gain factor $g$ that depends on $\sigma_x$ and $\sigma_n$.

Answer the following questions for part 1:

1. What is the form of $g(\sigma_x, \sigma_n)$?

2. Plot $g$ as a function of the signal to noise ratio, $\sigma_x^2/\sigma_n^2$, over the range .01 to 100 (use semilogx).

3. Explain what you see—i.e., how does the signal-to-noise ratio affect the relation between $\hat{x}$ and $y$?

## 2 Motion estimation

In the above example we saw how to use Bayesian inference to infer an underlying variable $x$ by combining a noisy measurement $y$ together with our prior knowledge of
the variable and the noise. Here, we will similarly estimate the velocity of a moving stimulus, \( v \), from noisy image pixel values \( I \). This procedure follows the same basic form of what we did above—its just a bit more complicated now because we are inferring a two-dimensional variable from many pixel observations, and the model is a bit more elaborate too.

First we need to specify the likelihood, \( P(I|v) \). As described in the methods section of the paper, this has the following form:

\[
P(I|v) \propto e^{-\frac{1}{2\sigma_n^2} \sum_{x,y} [I_x(x,y) v_x + I_y(x,y) v_y + I_t(x,y)]^2}
\]  

(11)

where \( I_x \) and \( I_y \) are spatial-derivatives of the image, \( I_t \) is the temporal derivative, and \( v = [v_x \ v_y] \) is the velocity.

The prior \( P(v) \) is isotropic, Gaussian and favors slow speeds (zero mean):

\[
P(v) \propto e^{-\frac{1}{2\sigma_p^2} [v_x^2+v_y^2]}
\]

(12)

\( \sigma_p \) determines the spread of the distribution.

Thus, the log-posterior has the following form:

\[
\log P(v|I) = -\frac{1}{2\sigma_n^2} \sum_{x,y} [I_x(x,y) v_x + I_y(x,y) v_y + I_t(x,y)]^2 - \frac{1}{2\sigma_p^2} [v_x^2 + v_y^2] + \text{const.}
\]

(13)

Differentiating with respect to \( v_x \) and \( v_y \) and setting the result to zero yields the solution shown in equation 1 of the Weiss et al. paper. Thus, \( v \) is essentially a function of the image derivatives, in addition to the ratio \( \sigma_p^2/\sigma_n^2 \) (it’s not the signal-to-noise ratio, but rather the dynamic range of speeds relative to the pixel noise). When \( \sigma_n \) is small, then the estimate of velocity is tied more directly to the measurements, \( I \) (first term in eq. 13). When \( \sigma_n \) is large then the velocity estimate will be more biased toward zero by the prior (second term in eq. 13). Note that lowering the contrast essentially has the same effect as increasing \( \sigma_n \), since the derivatives \( I_x, I_y, \) and \( I_t \) are smaller and hence the first term has less influence on \( v \).

Now let’s put all this into practice using the famous motion-plaid stimulus. A plaid pattern is generated by superimposing two sine-wave gratings of different orientations. You can get a Matlab script for generating moving plaid patterns from the course web page (plails.m). You will see that it has variables controlling the contrast, spatial-frequency, orientation, and other parameters of the component sine-wave gratings. Try playing around with these parameters, especially the relative contrast between the gratings. When the contrast is roughly equal you will see the combined motion of the plaid pattern, but when the contrast of one grating is low relative to the other then you should see them break up into two different patterns, or else you will get a single, combined motion percept that is biased in the direction of the grating with stronger contrast.

The script infer_v.m will compute the estimated velocity using the formula derived in Weiss et al. It simply computes the spatial and temporal derivatives of the image pixels and combines these into a motion estimate using equation 1 from the paper. You should be able to recreate figure 5b from the paper by calling this script within a for-loop that progressively decreases the contrast of the second grating:
for i=1:10
    c(i)=c1*2.^(-(i-1)/2);
    c2=c(i);
    infer_v
    b(i)=asin(v(1)/norm(v));
end

plot(log2(c1./c),b*180/pi)

Turn in the following for part 2:

1. A plot with $c_1=0.25$.
2. A plot with $c_1=0.1$ (superimposed on the above).
3. A qualitative/intuitive explanation of why this is happening, drawing upon the theory above.