Homework 1. CSE 598C Vision-based Tracking, Fall 2012. Due Wed Fri 7, 2012 in Angel.

This homework will actually be pretty easy, I think, if you followed the example on Gaussian Point Observations that we went over in class. You can view that example as one step in a recursive Bayes filter for estimating state x given a single observation y and a prior estimate mu. For homework, I'd like you to try two things:

1) consider a whole list of N observations y1,y2,...,yN and compute the mean value of the posterior you would get by combining all these observations with the prior. You could do this in at least two ways. You could form a joint likelihood from all N observations as a batch, and combine with the prior to get a joint posterior. Or you could starting cranking through recursively by taking y1 and combining with the prior to a get a posterior (this is exactly what we already did in the example in class), then taking that posterior as a new prior for a new step, to be combined with observation y2, to get a new posterior, and so on. It will only take two steps or so to discover the recursion relation for the mean and variance of the posterior after having seen $N > 1$ observations.

hint: the derivations may be easier if you write things in terms of "precision" (1 over variance) rather than variance. That is, if $p = 1/\sigma^2$ then we can write a 1D normal distribution with mean mu and standard deviation sigma as

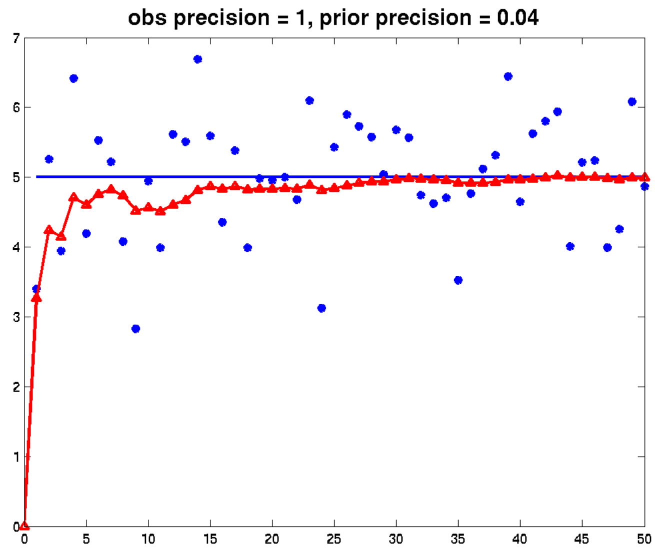
$$N(x|\mu, p) = c\ \exp\left\{-\frac{1}{2}p\,(x-\mu)^2\right\}$$

See my course notes on "completing the square" for more details about writing Gaussians in terms of precision instead of variance.

2) write a small matlab program to implement a recursive Bayes filter as in the above example, but by really computing the results one step at a time, iteratively, using the posterior computed after seeing observation y_t as a prior for the next time step, where you then will be incorporating observation y_{t+1}, and so on. In other words, you will be maintaining an estimate of the value of the current mean_t and current variance_t (or precision_t), and those estimates will keep getting updated iteratively as each new observation y_{t+1} comes in. Specifically, assume that each y_t is a noisy observation of a constant state value x, and for simplicity, that these observations are independent, and have the same variance. That is, use the model

$$\begin{aligned} p(x) &= N(\mu, \sigma^2) && \text{; prior distribution} \\ y_t &= x + \varepsilon && \text{; observation at time t} \\ p(\varepsilon) &= N(0, s^2) && \text{; observation noise} \end{aligned}$$

Finally, plot the results in a graph showing time as the x axis and the observations and estimated state values (mean of posterior distribution at time t) on the y axis, as in the figure below.



In this plot, the blue line is the true state value, blue points are the noisy observations, and red line is the estimated state value over time. This was produced for a true state value of 5, a prior distribution with mean 0 and precision 0.04 (variance 25), observation noise with mean 0 and precision 1 (variance 1), and running for 50 time steps (so there were 50 observations). Try playing around with the variance of the prior estimate. What happens if you set it low (that is, you mistakenly think you have a good prior)? Also, if you are ambitious, change the example so that instead of a constant state x that we are trying to estimate, that the state x also changes as a function of time. Say, for example, that $x_t = x_0 + t$. Does this same recursive estimator, which implicitly assumes that x is stationary, still do something reasonable?