2. Lectures 11, 12: Elements of probability and statistics

Plan:

- ◇ Parametric estimation continued
- ◇ Gaussian from binomial
- ◇ Multivariate Gaussian
- ◇ Covariance matrices and joint probabilities
- ◇ Weighted least squares
- ◇ Recursive least squares and the Kalman filter
- ◇ Data assimilation

2.1. **Parametric estimation continued.** Suppose, we have the data y_1, y_2, ..., y_n measured at points x_1, x_2, ..., x_n. We want to fit the data to some curve $y = f(x, \alpha)$ with parameters α_1, α_2, ..., α_m, $m < n$. We assume that the data y_i have Gaussian distribution about the values $f(x_i, \alpha)$ with some known variance σ^2. Then, the probability of getting the n values of y_i is

$$P(y) = \prod_{i=1}^{n} P(y_i, x_i, \sigma) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left[-\sum_{i=1}^{n} \frac{(y_i - f(x_i, \alpha))^2}{2\sigma^2}\right].$$

To maximize the likelihood function P, we minimize the argument of the exponential. With σ known, we need to minimize the square error. In general, this is the nonlinear problem.

We now assume that $f = \alpha^T g(x)$ with some functions $g_i(x)$ in vector g. The problem becomes linear least squares problem to find α.

The wish equation is $G\alpha = y$, and the least squares solution is given by the pseudo-inverse as

$$\alpha = G^+ y,$$

where

$$G^+ = \left(G^T G\right)^{-1} G^T.$$

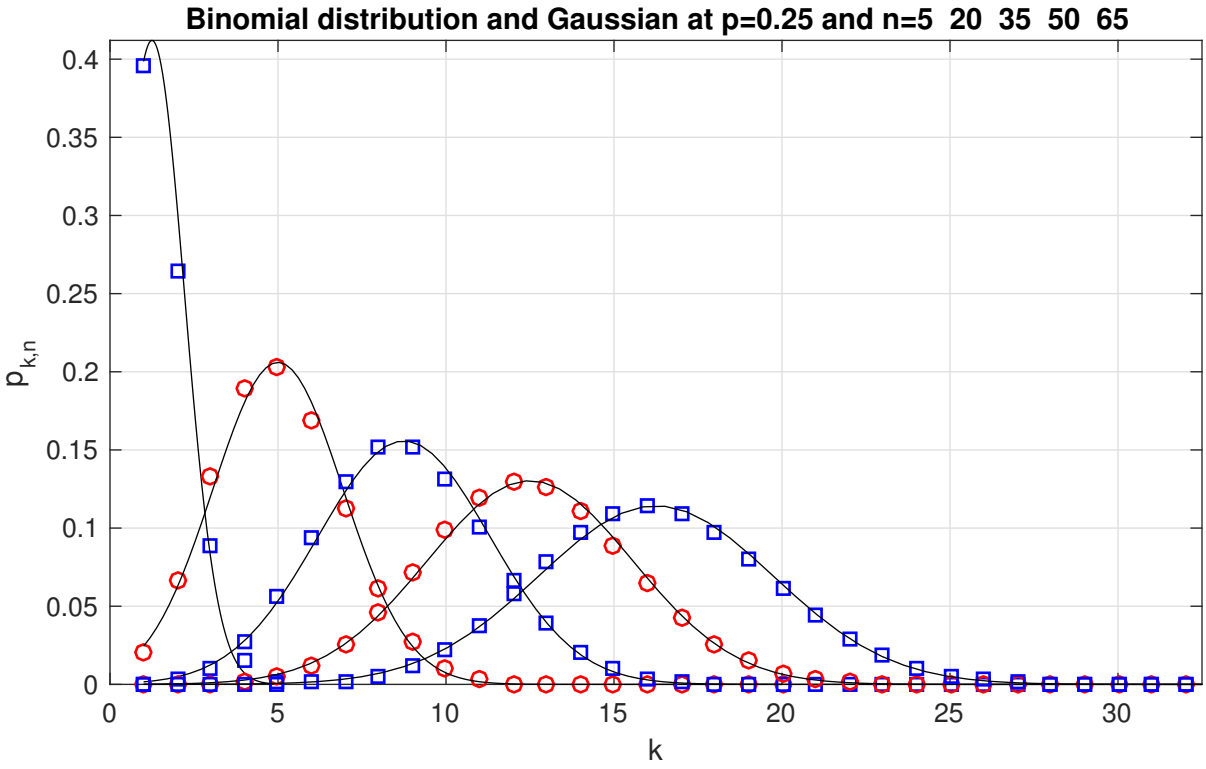

FIGURE 2.1. Binomial probability of success at increasing number of trials.

2.2. **Gaussian from binomial.** To illustrate the central limit theorem, we look at the behavior of the binomial distribution when the number of trials becomes large. We have

$$p_{k,n} = C_k^n p^k (1-p)^{n-k}, \quad C_k^n = \frac{n!}{k!\,(n-k)!}$$

as the probability of success in k flips out of n total. Every flip has probability of success p.

Now we plot $p_{k,n}$ as a function of k for increasing values of n, Fig. 2.1. The plot is compared with the Gaussian

$$f = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

where $\mu = np$ and $\sigma^2 = np\,(1-p)$ are the mean and variance of the binomial distribution as derived earlier. We can see that very fast, with increasing number of trials n, the binomial distribution becomes close to the Gaussian in agreement with the central limit theorem.

2.3. **Multivariate Gaussians.** Let x and y be Gaussian random variables with means m_1 and m_2 and variances σ_1^2 and σ_2^2. If they are independent, then

$$p(x,y) = p(x)\,p(y) = \frac{1}{2\pi\sigma_1\sigma_2}\exp\left[-\frac{(x-m_1)^2}{2\sigma_1^2}\right]\exp\left[-\frac{(y-m_2)^2}{2\sigma_2^2}\right]$$

and $\sigma_{12} = 0$ resulting in the diagonal covariance matrix

$$V = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}.$$

Note that the exponents can be combined into

$$-\frac{1}{2}(x-m)^T V^{-1}(x-m).$$

Now generalize to more variables and drop independence:

$$p(x) = \frac{1}{\left(\sqrt{2\pi}\right)^M \sqrt{\det(V)}}\exp\left[-\frac{1}{2}(x-m)^T V^{-1}(x-m)\right],$$

where the covariance matrix can now be full, but still symmetric. One can use the spectral decomposition of V and normalize to show where the determinant in front comes from.

2.4. **Covariance matrices and joint probabilities.** We run n experiments at once and measure n values x_1, x_2, ..., x_n. The measurements x_i may or may not be independent. Every measured x_i will have some error e_i to it.

Let $\mu_i = x_i + e_i$ be the measurement mean.

If the errors e_i and e_j are independent, then $e_i e_j$ has mean zero.

The average of $e_i e_j$ is the *covariance*

$$\sigma_{ij} = \sum_{\text{over realizations}} p_{ij} e_i e_j = E\left[e_i e_j\right],$$

where p_{ij} are joint probabilities of errors in measurements i and j.

The covariance matrix is defined as

$$V = (\sigma_{ij})$$

and it will be diagonal when e_i and e_j are independent.

The diagonal element σ_{ii} of V is the *variance* σ_i^2 of the errors in i-th measurement.

If we define the error vector of the experiment as

$$e = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix},$$

then the covariance matrix can be written as

$$V = E\left[ee^T\right] = E\begin{bmatrix} e_1^2 & e_1 e_2 & \cdots & e_1 e_n \\ e_2 e_1 & e_2^2 & \cdots & e_2 e_n \\ & & \vdots & \\ e_n e_1 & e_n e_2 & \cdots & e_n^2 \end{bmatrix}.$$

If one runs the experiment N times to collect the data

$$\begin{aligned} x^1 &= \left(x_1^1, x_2^1, ..., x_n^1\right) \\ x^2 &= \left(x_1^2, x_2^2, ..., x_n^2\right) \\ &\cdots \\ x^N &= \left(x_1^N, x_2^N, ..., x_n^N\right), \end{aligned}$$

then the sample mean is

$$\bar{\mu}_i = \frac{1}{N}\sum_{k=1}^{N} x_i^k$$

and the sample variance

$$\bar{\sigma}_{ij} = \frac{1}{N-1}\sum_k \left(x_i^k - \bar{\mu}_i\right)\left(x_j^k - \bar{\mu}_j\right).$$

Again, the sample variance is diagonal if the measurements are independent.

The dimensionless quantity

$$\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j}$$

is called the *correlation coefficient*.

2.5. **Weighted least squares.** We return to the least-squares problem, $Ax = b$, and ask a question:

- ◇ If we have some error in b, with the covariance matrix V, what is the corresponding error in the least-squares solution u of $A^TAu = A^Tb$?
- ◇ In other words, we want to estimate the *reliability* of such a solution. This reliability is measured by the covariance matrix W of the error in u. Next, we want to derive an expression for W.

Since not all equations in $Ax = b$ are equivalent in terms of the error they introduce into the solution, we divide these equations by various coefficients that are supposed to give *different weights to different equations*. Of course, at this point we have no clue which equations should be given higher weight, which lower.

The rescaling is provided by multiplication with a matrix C, to be found:

$$CAx = Cb.$$

Now we form the normal equation:

$$A^TCAu = A^TCb,$$

from which we get

$$u = \left(A^TCA\right)^{-1} A^TCb = Lb$$

with matrix L defined as

$$L = \left(A^TCA\right)^{-1} A^TC.$$

Note that

$$LA = I.$$

We want covariances for the output error

$$x - u.$$

Using $u = Lb$ and $x = Ix = LAx$, we get

$$x - u = LAx - Lb = L\left(Ax - b\right) = -Le,$$

so this is the relationship between the error in the right-hand side b and the error in the solution $x - u$.

Next, the required covariance is

$$\begin{aligned} W &= E\left[\left(x-u\right)\left(x-u\right)^T\right] = \\ &= E\left[Lee^TL^T\right] = \\ &= LE\left[ee^T\right]L^T = \\ &= LVL^T. \end{aligned}$$

We took L outside of the expectation assuming it is constant.

The next step is to minimize this W by choosing appropriate C in L.

Theorem 11. *The best W is obtained with $C = V^{-1}$, in which case $W = \left(A^TV^{-1}A\right)^{-1}$.*

Proof. To verify, just plug in $C = V^{-1}$ into $L = \left(A^TCA\right)^{-1} A^TC$ to obtain

$$L^* = \left(A^TV^{-1}A\right)^{-1} A^TV^{-1}$$

and then compute LVL^T.

If we take a different choice of C it will change L^* to $L = L^* + (L - L^*)$, but then W will be bigger, in the sense that a positive semi-definite matrix will be added. Indeed,

$$W = LVL^T = L^*VL^{*T} + (L - L^*)VL^{*T} + L^*V(L - L^*)^T + (L - L^*)V(L - L^*)^T.$$

The two middle terms (transposes of each other) can be shown to be zero. Indeed

$$(L-L^*)VL^{*T} = (L-L^*)V\ V^{-1}A\left(A^TV^{-1}A\right)^{-1} = (LA-L^*A)\left(A^TV^{-1}A\right)^{-1} = 0$$

since $LA = 0$ and $L^*A = 0$ as any L must satisfy $LA = I$ according to its definition above.

The last term in W, $(L-L^*)V(L-L^*)^T$, is positive semi-definite, and therefore W is smallest when this term vanishes, i.e. when $L = L^*$. □

Matrix

$$W^{-1} = A^TV^{-1}A$$

is called the *information matrix* since:

- ◇ When V is small, i.e. variances are small (hence better accuracy), this matrix W^{-1} is large, it contains more information.
- ◇ Furthermore, W^{-1} also increases, when more data are added that increase the matrix A with more rows.

Note that W does not depend on b, only on the error in b and matrix A.

Example 12. Suppose a doctor measures your heart rate x three times ($m = 3$, $n = 1$). Then $Ax = b$ with

$$A = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad b = \begin{bmatrix} 60 \\ 70 \\ 90 \end{bmatrix}, \quad \text{so } x = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 60 \\ 70 \\ 90 \end{bmatrix}.$$

And assume that $V = diag\begin{pmatrix} \sigma_1^2, & \sigma_2^2, & \sigma_3^2 \end{pmatrix} = diag\begin{pmatrix} \frac{1}{9}, & \frac{1}{4}, & 1 \end{pmatrix}$, which implies that with every measurement, the error has increased. Something went wrong between you and the doctor. We want the best estimate of the heart rate.

You could of course just ignore the second and third measurements and decide that the heart rate is 60. This would mean that you have put zero weight to those equations. But perhaps it still makes sense to account for them to improve the estimate, even though giving them smaller significance.

This is what weighted lease squares does, which is obtained by solving the normal equation $A^TV^{-1}Au = A^TV^{-1}b$:

$$\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 9 & & \\ & 4 & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} u = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 9 & & \\ & 4 & \\ & & 1 \end{bmatrix} \begin{bmatrix} 60 \\ 70 \\ 90 \end{bmatrix}.$$

Solving this, we get

$$u = \frac{9b_1 + 4b_2 + b_3}{14} = 65$$

as the best weighted average.

Then, the variance of u is

$$W = \left(A^TV^{-1}A\right)^{-1} = \left(\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 9 & & \\ & 4 & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right)^{-1} = \frac{1}{14},$$

which is smaller than $\frac{1}{9}$ in the first measurement b_1 as we have accounted for the additional measurements that helped improve the result.

2.6. **Recursive least squares and the Kalman filter.** Start with a simple example.

Example 13. Given the data b_1, b_2, ..., b_{99}, we find the average as

$$\hat{u}_0 = \frac{1}{99}\left(b_1 + b_2 + ... + b_{99}\right).$$

Now we have one more data point, b_{100}. How do we find the new average?

There is a dumb way to do it:

$$\hat{u}_1 = \frac{1}{100}\left(b_1 + b_2 + ... + b_{100}\right),$$

requiring to recalculate the whole sum again.

And there is a better way that reuses the previous calculation and finds the new average as the old average plus some correction:

$$\begin{aligned}
\hat{u}_1 &= \frac{1}{100}\left(b_1 + b_2 + ... + b_{100}\right) = \\
&= \frac{99}{100}\frac{1}{99}\left(b_1 + b_2 + ... + b_{99}\right) + \frac{1}{100}b_{100} = \\
&= \frac{99}{100}\hat{u}_0 + \frac{1}{100}b_{100} = \\
&= \hat{u}_0 + \underbrace{\underbrace{\frac{1}{100}}_{\text{gain factor}}\underbrace{\left(b_{100} - \hat{u}_0\right)}_{\text{innovation}}}_{\text{update}}.
\end{aligned}$$

This example could be put in the language of least squares for $Ax = b$ by letting

$$A_0 = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \quad b_0 = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{99} \end{bmatrix},$$

so that

$$\begin{aligned}
A_0^T A_0 \hat{u}_0 &= A_0^T b_0 \\
99\hat{u}_0 = b_1 + b_2 + ... + b_{99}, &\to \hat{u}_0 = \frac{1}{99}\left(b_1 + b_2 + ... + b_{99}\right).
\end{aligned}$$

With the new data in A and b, the equation becomes

$$Ax = \begin{bmatrix} A_0 \\ A_1 \end{bmatrix} x = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix},$$

where the matrix A now has new data in A_1 and b has new data in b_1. The least squares solution for this new system is given by the normal equation

$$\begin{bmatrix} A_0^T & A_1^T \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \end{bmatrix} \hat{u}_1 = \begin{bmatrix} A_0^T & A_1^T \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix},$$

for which we also want to get the result that reuses the old calculation.

Multiplying through the matrices in the normal equation we get

$$\begin{aligned} A^T A &= A_0^T A_0 + A_1^T A_1 \\ A^T b &= A_0^T b_0 + A_1^T b_1 = \\ &= A_0^T A_0 \hat{u}_0 + A_1^T b_1 = \\ &= \left(A^T A - A_1^T A_1\right) \hat{u}_0 + A_1^T b_1, \end{aligned}$$

hence

$$\begin{aligned} \hat{u}_1 &= \left(A^T A\right)^{-1} A^T b = \\ &= \hat{u}_0 + \left(A^T A\right)^{-1} \left(-A_1^T A_1 \hat{u}_0 + A_1^T b_1\right) = \\ &= \hat{u}_0 + \underbrace{\left(A^T A\right)^{-1} A_1^T}_{\text{Kalman gain matrix}} \underbrace{\left(b_1 - A_1 \hat{u}_0\right)}_{\text{innovation}} = \\ &= \hat{u}_0 + K_1 \left(b_1 - A_1 \hat{u}_0\right). \end{aligned}$$

For our example, $A_1 = [1]$ is 1×1 matrix, $b_1 = [1]$ is 1×1 vector, and we get $A^T A = 100$, and $K_1 = 1/100$.

Now more generally, we may have a problem that gets more complex data in, and the least squares problem may involve weights. How do we update then? We follow a similar path.

Let

$$\begin{aligned} A_0 x_0 &= b_0, \quad \text{old problem.} \\ A_0^T V_0^{-1} A_0 \hat{u}_0 &= A_0^T V_0^{-1} b_0, \quad \text{least squares solution} \end{aligned}$$

Here V_0 is the covariance matrix of errors in b_0.

With the new data coming in A_1 and b_1, we need to solve a new system:

$$\begin{bmatrix} A_0 \\ A_1 \end{bmatrix} x_1 = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}, \quad \text{new problem.}$$

Then the normal equation is

$$\begin{bmatrix} A_0^T & A_1^T \end{bmatrix} \begin{bmatrix} V_0^{-1} & \\ & V_1^{-1} \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \end{bmatrix} \hat{u}_1 = \begin{bmatrix} A_0^T & A_1^T \end{bmatrix} \begin{bmatrix} V_0^{-1} & \\ & V_1^{-1} \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}.$$

Here V_1 is the extra part of the covariance matrix V due to the new data in b. We want to solve this equation as before with the new $\hat{u}_1$ found as an update to $\hat{u}_0$:

$$\hat{u}_1 = \hat{u}_0 + K_1 \left(b_1 - A_1 \hat{u}_0\right)$$

with the gain matrix K_1 and the innovation $b_1 - A_1 \hat{u}_0$, the mismatch between the previous state $A_1 \hat{u}_0$ and the new measurement b_1.

To find the Kalman gain matrix K_1, we expand the normal equation for $\hat{u}_1$.

$$\begin{bmatrix} A_0^T & A_1^T \end{bmatrix} \begin{bmatrix} V_0^{-1}A_0 \\ V_1^{-1}A_1 \end{bmatrix} \hat{u}_1 = \begin{bmatrix} A_0^T V_0^{-1} & A_1^T V_1^{-1} \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix},$$

$$\left[A_0^T V_0^{-1} A_0 + A_1^T V_1^{-1} A_1\right] \hat{u}_1 = \begin{bmatrix} A_0^T V_0^{-1} b_0 + A_1^T V_1^{-1} b_1 \end{bmatrix} =$$

$$\underbrace{\left[\underbrace{A_0^T V_0^{-1} A_0}_{W_0^{-1}} + A_1^T V_1^{-1} A_1 \right]}_{W_1^{-1}} \hat{u}_1 = \begin{bmatrix} \underbrace{A_0^T V_0^{-1} A_0}_{W_0^{-1}} \hat{u}_0 + A_1^T V_1^{-1} b_1 \end{bmatrix}.$$

Then

$$\begin{aligned} \hat{u}_1 &= W_1 \left[\left(W_1^{-1} - A_1^T V_1^{-1} A_1\right) \hat{u}_0 + A_1^T V_1^{-1} b_1 \right] = \\ &= \hat{u}_0 - W_1 A_1^T V_1^{-1} A_1 \hat{u}_0 + W_1 A_1^T V_1^{-1} b_1 = \\ &= \hat{u}_0 + \underbrace{W_1 A_1^T V_1^{-1}}_{K_1} \left(b_1 - A_1 \hat{u}_0\right). \end{aligned}$$

Hence

$$K_1 = W_1 A_1^T V_1^{-1},$$

where the covariance of errors in u_1 is found from

$$W_1^{-1} = W_0^{-1} + A_1^T V_1^{-1} A_1.$$

The old covariance matrix was

$$W_0 = \left(A_0^T V_0^{-1} A_0\right)^{-1}.$$

To invert W_1^{-1} and get W_1, we may want to use the Sherman-Morrison-Woodbury formula for a matrix A that is corrected with something "smaller" (like a rank-1 matrix uv^T):

$$\left(A - UB^{-1}V\right)^{-1} = A^{-1} + A^{-1}U\left(B - VA^{-1}U\right)^{-1} VA^{-1}$$

Hence we find that,

$$\begin{aligned} W_1 &= \left(W_0^{-1} + A_1^T V_1^{-1} A_1\right)^{-1} = \\ &= W_0 - W_0 A_1^T \left(V_1 + A_1 W_0 A_1^T\right)^{-1} A_1 W_0, \end{aligned}$$

a formula that allows us to find W_1 without pain directly in terms of W_0 and the new data in A_1 and V_1.

2.7. **Data assimilation (follows Kutz, Ch. 21).** The goal is to improve a model by adding (assimilating) information from available data. For example, the model can be of the form

$$\begin{aligned} \frac{dx}{dt} &= f(x,t), \\ x(0) &= x_0. \end{aligned}$$

Generally, there will be some noise in the equation itself and in the initial data. Then

$$\begin{aligned} \frac{dx}{dt} &= f(x,t) + q_1, \\ x(0) &= x_0 + q_2. \end{aligned} \tag{2.1}$$

Both q_1 and q_2 will affect the solution and may do so quite significantly, especially when the system is sensitive to initial conditions, such as in chaotic systems. Then, we want to do some measurements that will give certain relationships between the data

$$g(x,t) + q_3 = 0 \tag{2.2}$$

with their own errors in q_3. And we want to somehow use this extra information to improve the predictions of the main dynamical system. This is the idea of data assimilation, and it will be illustrated on the Lorenz system:

$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= rx - y - xz \\ \dot{z} &= xy - bz. \end{aligned}$$

Adding the constraints (2.2) overdetermines the system. We introduce a functional of the errors

$$J(x) = \int\int dt_1 dt_2 q_1^T(t_1) C_1 q_1(t_2) + q_2^T C_2 q_2 + q_3^T C_3 q_3$$

which includes as weights the inverses of the covariance matrices of the errors in the equation, q_1, the initial conditions, q_2, and the measurements, q_3.

So, we want to find such dynamics $x(t)$ that minimizes the error functional $J(x)$. This minimization of the quadratic J can be turned into the maximum likelihood problem by requiring that the probability density function built from J, namely

$$P(x) = c\exp(-J(x)),$$

has a maximum.

Data assimilation for single variable. Suppose the model predicts x with some error, and measurement makes its own prediction with another error. We want to combine these predictions to improve the x.

If y is the observation, x is the model prediction, then the probability of finding x given the data observation is, by Bayes,

$$p(x|y) = \frac{p(y|x)\,p(x)}{p(y)} \tag{2.3}$$

with:

$p(x)$ - the probability of finding x;

$p(y|x)$ - the likelihood function.

We will now turn to Gaussian distributions for p:

$$p(y|x) = c_1 \exp\left[-\frac{(y-x)^2}{2\sigma_y^2}\right]$$
$$p(x) = c_2 \exp\left[-\frac{(x-x_0)^2}{2\sigma_0^2}\right].$$

Where:

σ_y is the error variance of observation

x_0, σ_0 – predicted model mean and the error variance.

So, the model predicts x_0 as the most likely answer for x, with variance σ_0. We want to bring in the data with its own distribution $p(y|x)$ to improve the model prediction. This is done using (2.3) as

$$p(x|y) = c_3 \exp\left[-\frac{(y-x)^2}{2\sigma_y^2}\right] \exp\left[-\frac{(x-x_0)^2}{2\sigma_0^2}\right].$$

Maximizing this probability over x is equivalent to minimizing the quadratic form

$$\frac{(y-x)^2}{2\sigma_y^2} + \frac{(x-x_0)^2}{2\sigma_0^2}.$$

And this is accomplished by the choice

$$\bar{x} = \frac{\sigma_y^2}{\sigma_y^2+\sigma_0^2}x_0 + \frac{\sigma_0^2}{\sigma_y^2+\sigma_0^2}y,$$

which is a weighted sum of the model prediction x_0 and the measurement y.

Note that $\bar{x} = x_0$ if $\sigma_0 = 0$ and $\bar{x} = y$ if $\sigma_y = 0$, both of which are easy to understand.

The variance of error in x is

$$\bar{\sigma}^2 = \frac{\sigma_0^2\sigma_y^2}{\sigma_y^2+\sigma_0^2} \leq \sigma_0^2, \sigma_y^2,$$

hence combining both the model and observation makes things only better.

Now we put on the Kalman glasses and look at the formula for $\bar{x}$ differently:

$$\begin{aligned}\bar{x} &= \frac{\sigma_y^2}{\sigma_y^2+\sigma_0^2}x_0 + \frac{\sigma_0^2}{\sigma_y^2+\sigma_0^2}y = \\ &= x_0 + \frac{\sigma_0^2}{\sigma_y^2+\sigma_0^2}(y-x_0) = \\ &= x_0 + \underbrace{K}_{\text{Kalman gain factor}}\underbrace{(y-x_0)}_{\text{innovation}}.\end{aligned}$$

We improve on x_0 by adding a correction $K(y-x_0)$ due to observations.

Next, we want to generalize these observations to systems.

Data assimilation for systems. Note now that in systems we may have measurements only at few points, while the model predicts much more. So we need to a way of mapping the observations y to the state vector x. Assuming a linear map, this will be carried out via matrix H as follows:

$$y(t) = Hx(t) + q_3.$$

Thus, the state vector $x(t)$ at time t is mapped to observations $y(t)$, and there is some error q_3.

Suppose now that we have the true dynamics given by

$$x_{k+1} = f(x_k) + q_{k+1}$$

which maps the state x_k at time t_k to state x_{k+1} at time t_{k+1}. We will assume q_{k+1} is the Gaussian noise.

The model approximation is given by

$$x_{0k+1} = f(x_{0k}),$$

where x_{0k} is the best estimate of the state of the system at t_k and x_{0k+1} is the forecast.

The error between the true state and forecast is

$$x_{k+1} - x_{0k+1} = f(x_k) - f(x_{0k}) + q_{k+1}$$

and we can approximate this by Taylor expansion as

$$x_{k+1} - x_{0k+1} = f'(x_{0k})(x_k - x_{0k}) + ... + q_{k+1}.$$

The error variance is then

$$E\left[(x_{k+1} - x_{0k+1})^2\right] = \left(f'(x_{0k})\right)^2 E\left[(x_k - x_{0k})^2\right] + E\left[q_{k+1}^2\right] + ...$$

Next, define

$$P_{k+1} = E\left[(x_{k+1} - x_{0k+1})^2\right]$$
$$P_k = E\left[(x_k - x_{0k})^2\right].$$

Then

$$P_{k+1} = \left(f'(x_{0k})\right)^2 P_k + E\left[q_{k+1}^2\right].$$

The prediction accounting for data assimilation is then, similarly to the earlier example above,

$$\bar{x}_{k+1} = x_{0k+1} + K_{k+1}(y_{k+1} - x_{0k+1}),$$

where

$$K_{k+1} = \frac{P_{k+1}}{P_{k+1} + R}$$

with R the variance of the observation error. Note, if $R = 0$, i.e. the observation is perfect, then $K_{k+1} = 1$ and hence $\bar{x}_{k+1} = y_{k+1}$, the prediction is also perfect.

The vector version of this calculation is similar.

$$x_{k+1} = f(x_k) + q_{k+1}$$
$$x_{0k+1} = f(x_{0k}).$$

Then

$$P_{k+1} = J(f) P_k J^T(f) + Q$$

where J is the Jacobian in the Taylor expansion. Then

$$\bar{x}_{k+1} = x_{0k+1} + K_{k+1}(y_{k+1} - Hx_{0k+1}),$$

and the Kalman matrix is given by

$$K_{k+1} = P_{k+1} H^T \left(H P_{k+1} H^T + R\right)^{-1}$$

Data assimilation for the Lorenz system. We now look at the Lorenz system

$$\begin{aligned} \dot{x} &= \sigma (y - x) \\ \dot{y} &= rx - y - xz \\ \dot{z} &= xy - bz \end{aligned} \tag{2.4}$$

with $\sigma = 10$, $r = 8/3$, $b = 28$. The IC is $[5, 5, 5]$ or this with an added noise. The system is solved over t from 0 to 20.

We consider adding noise to the IC, σ_2, and to the measurements, σ_3, but not to the equations.

The measurements are for simplicity taken from noise-free solution with some added noise

$$\mathrm{y}(t_n) = \mathrm{x}(t_n) + \sigma_3 q(0, 1)$$

with t_n the measurement points, $q(0, 1)$ is Gaussian with mean zero and unit variance.

With $H = I$, the Kalman gain matrix is

$$K_{k+1} = P_{k+1} (P_{k+1} + R)^{-1}.$$

Since the noise σ_2 and σ_3 is the same for all equations in our example, we will have

$$K_{k+1} = \frac{\sigma_2}{\sigma_2 + \sigma_3} I.$$

Since $q_1 = 0$, we also have

$$P_{k+1} = J P_k J^T,$$

where

$$J = \begin{bmatrix} -\sigma & \sigma & 0 \\ r - z & -1 & -x \\ y & x & -b \end{bmatrix}.$$

The Matlab code to do the calculations is below and its output is shown in the figures that follow.

```
function data_assimilation_lorenz(sigma,b,r)
% data assimilation for Lorenz system. Based on Kutz, Ch. 21

% default arguments if no input
if nargin < 1,
    sigma = 10;      b=8/3;       r=28;
end

t=0:0.01:20;
x0=[5 5 5];

% true solution of the Lorenz system with the above IC
[t,xsol] = ode45(@lor_rhs,t,x0);
x_true=xsol(:,1); y_true=xsol(:,2); z_true=xsol(:,3);
figure(1), plot3(x_true,y_true,z_true), grid on %the attractor
xlabel x, ylabel y, zlabel z


%plot the solution with different noise in the IC to show sensitivity to IC
figure(2)
sigma2=1; % variance for error in the IC
```

```
for j=1:8
    xic = x0 + sigma2*randn(1,3);   % perturb IC by Gaussian random noise
    [t,xsol]=ode45(@lor_rhs,t,xic);
    x = xsol(:,1);                  % projected x values
    y = xsol(:,2);                  % projected y values
    z = xsol(:,3);                  % projected z values
    subplot(4,2,j), plot(t,x_true,'k')  %true solution
    hold on
    plot(t,x,'b','Linewidth',1)     % solution from noisy IC
end


% now collect observations every 50 time steps, i.e. every t=0.5 time units
% these will be taken as the true solution plus some random noise

tdata=t(1:50:end);
n = length(tdata);
xn = randn(n,1);
yn = randn(n,1);
zn = randn(n,1);
sigma3 = 2;                                 % error variance in measured data
xdata = x_true(1:50:end) + sigma3*xn;   % measured data
ydata = y_true(1:50:end) + sigma3*yn;
zdata = z_true(1:50:end) + sigma3*zn;

figure(3)  %this shows the true solution and the measured data points
plot(t,x_true,'k',tdata,xdata,'go','Linewidth',2)

figure(4)
%the main data assimilation part, where the measurements will
%be included to force the solution to stay close to the measured points
%every 0.5 time units, a new data point is picked that is used to adjust
%the solution at the end of this interval, and this updated solution is
%used as an IC for the next interval, and so on. We move forward picking up
%data on the fly and using them to correct the solution.

x_da=[];                                % data assimilation solution
for j=1:length(tdata)-1                 % step through every t=0.5
    tspan = 0:0.01:0.5;                 % time between data collections,[0,0.5]
    [tspan,xsol] = ode45(@lor_rhs,tspan,xic);
    xic0 = [xsol(end,1); xsol(end,2); xsol(end,3)]; % model estimate
    xdat = [xdata(j+1); ydata(j+1); zdata(j+1)];    % data estimate
    K    = sigma2/(sigma2+sigma3);                  % Kalman gain
    xic  = xic0 + K*[xdat-xic0];                    % adjusted state vector
    x_da = [x_da; xsol(1:end-1,:)];                 % store the data
```

```
end

x_da=[x_da; xsol(end,:)]; % store last data time
plot(t,x_true,'k',t,x_da(:,1),'g','Linewidth',[2]) %true vs assimilated

% the right-hand side of the Lorenz system
function rhs = lor_rhs(t,x)
    rhs = [ sigma*(-x(1) + x(2))
            -x(1)*x(3)+r*x(1)-x(2)
            x(1)*x(2)-b*x(3)];
end


end
```

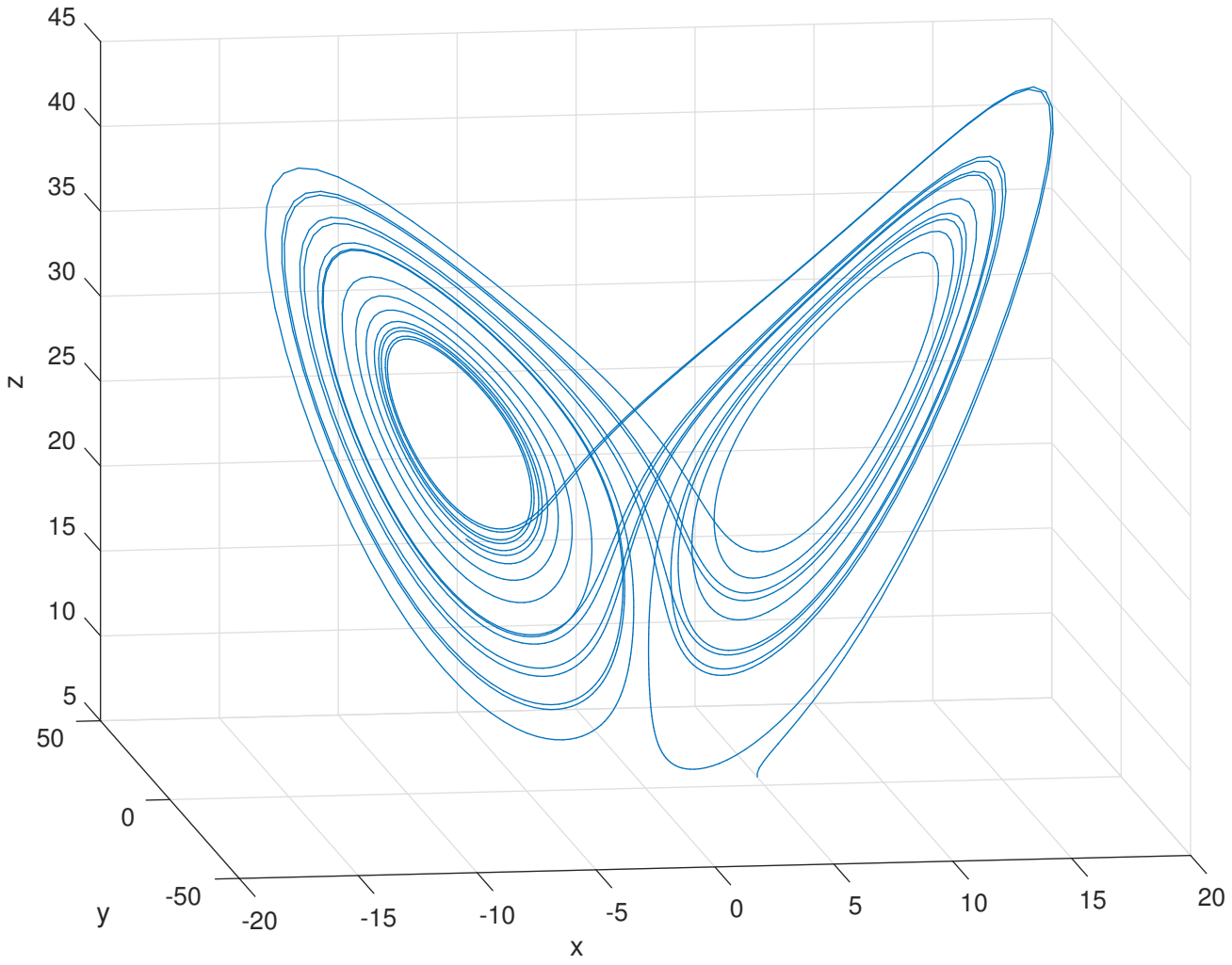

FIGURE 2.2. Figure(1): The Lorenz attractor.

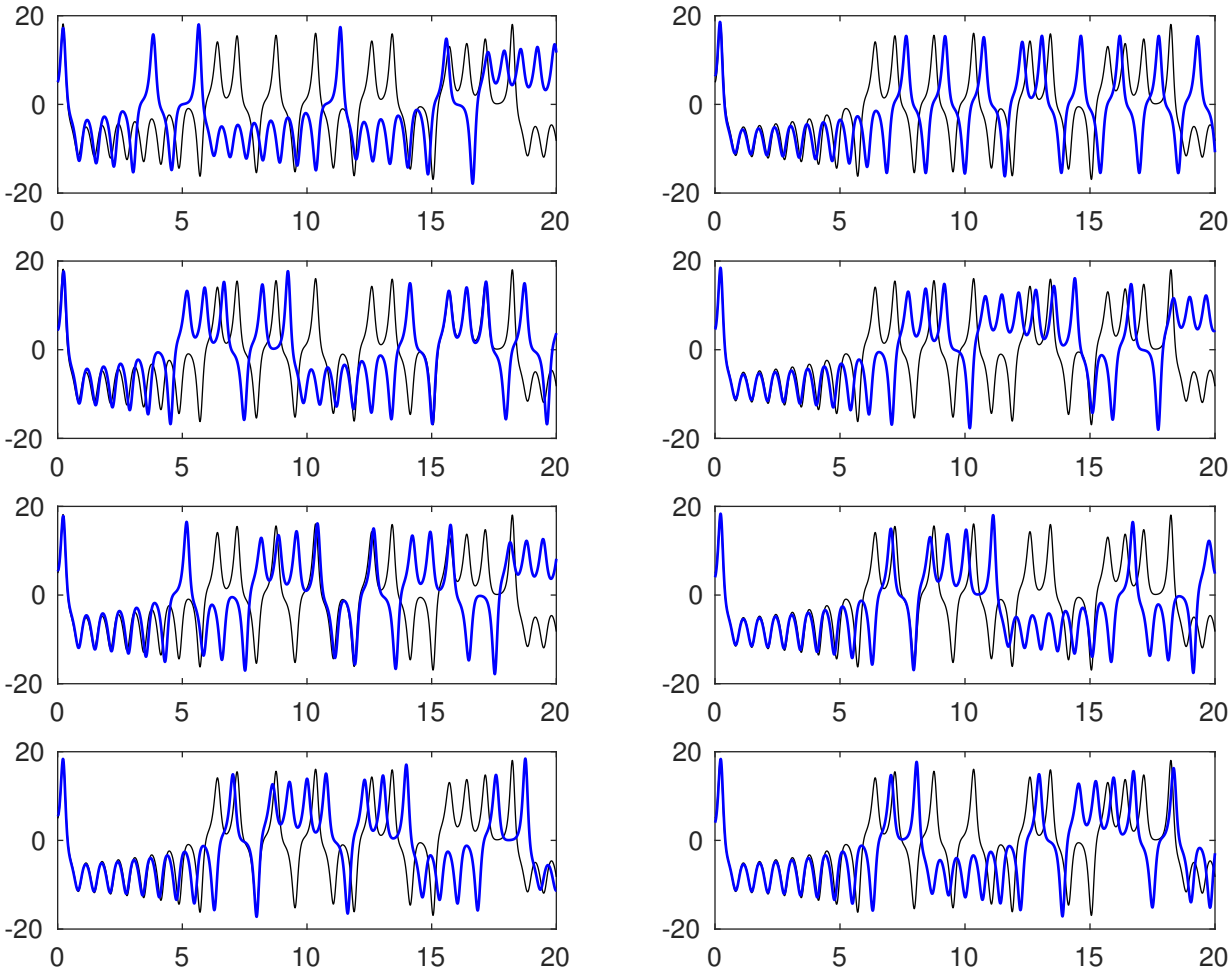

FIGURE 2.3. Figure(2): Solutions from noisy initial data.

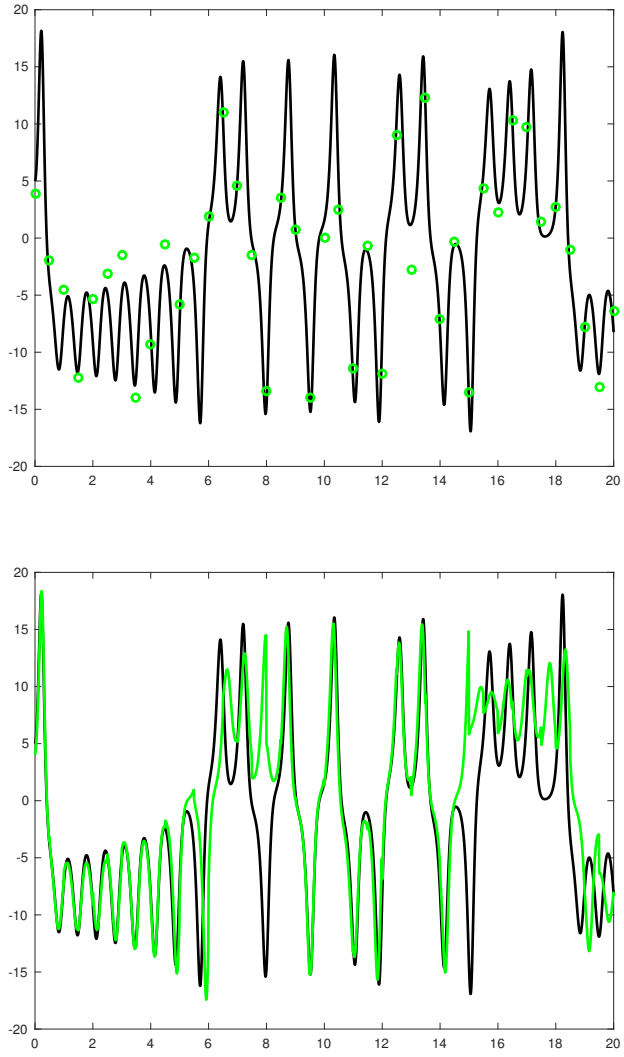

FIGURE 2.4. Figure(3): The measured data. Figure(4): The assimilated solution vs the true solution.