



1. *Fitting a Gaussian function to data.* A Gaussian function has the form

$$f(t) = ae^{-(t-\mu)^2/\sigma^2}.$$

Here $t \in \mathbf{R}$ is the independent variable, and $a \in \mathbf{R}$, $\mu \in \mathbf{R}$, and $\sigma \in \mathbf{R}$ are parameters that affect its shape. The parameter a is called the *amplitude* of the Gaussian, μ is called its *center*, and σ is called the *spread* or *width*. We can always take $\sigma > 0$. For convenience we define $p \in \mathbf{R}^3$ as the vector of the parameters, *i.e.*, $p = [a \ \mu \ \sigma]^T$.

We are given a set of data,

$$t_1, \dots, t_N, \quad y_1, \dots, y_N,$$

and our goal is to fit a Gaussian function to the data. We will measure the quality of the fit by the root-mean-square (RMS) fitting error, given by

$$E = \left(\frac{1}{N} \sum_{i=1}^N (f(t_i) - y_i)^2 \right)^{1/2}.$$

Note that E is a function of the parameters a , μ , σ , *i.e.*, p . Your job is to choose these parameters to minimize E . You'll use the Gauss-Newton method.

- (a) Work out the details of the Gauss-Newton method for this fitting problem. Explicitly describe the Gauss-Newton steps, including the matrices and vectors that come up. You can use the notation $\Delta p^{(k)} = [\Delta a^{(k)} \ \Delta \mu^{(k)} \ \Delta \sigma^{(k)}]^T$ to denote the update to the parameters, *i.e.*,

$$p^{(k+1)} = p^{(k)} + \Delta p^{(k)}.$$

(Here k denotes the k th iteration.)

- (b) Get the data t , y (and N) from the file `gauss_fit_data.m`, available on the class website. Implement the Gauss-Newton (as outlined in part (a) above). You'll need an initial guess for the parameters. You can visually estimate them (giving a short justification), or estimate them by any other method (but you must explain your method).

Plot the RMS error E as a function of the iteration number. (You should plot enough iterations to convince yourself that the algorithm has nearly converged.) Plot the final Gaussian function obtained along with the data on the same plot.

Repeat for another reasonable, but different initial guess for the parameters. Repeat for another set of parameters that is *not* reasonable, *i.e.*, not a good guess for the parameters. (It's possible, of course, that the Gauss-Newton algorithm doesn't converge, or fails at some step; if this occurs, say so.) Briefly comment on the results you obtain in the three cases.

2. *Spectral resolution of the identity.* Suppose $A \in \mathbf{R}^{n \times n}$ has n linearly independent eigenvectors p_1, \dots, p_n , $p_i^T p_i = 1$, $i = 1, \dots, n$, with associated eigenvalues λ_i . Let $P = [p_1 \ \dots \ p_n]$ and $Q = P^{-1}$. Let q_i^T be the i th row of Q .

- (a) Let $R_k = p_k q_k^T$. What is the range of R_k ? What is the rank of R_k ? Can you describe the null space of R_k ?
- (b) Show that $R_i R_j = 0$ for $i \neq j$. What is R_i^2 ?
- (c) Show that

$$(sI - A)^{-1} = \sum_{k=1}^n \frac{R_k}{s - \lambda_k}.$$

Note that this is a partial fraction expansion of $(sI - A)^{-1}$. For this reason the R_i 's are called the *residue* matrices of A .

- (d) Show that $R_1 + \dots + R_n = I$. For this reason the residue matrices are said to constitute a *resolution of the identity*.
- (e) Find the residue matrices for

$$A = \begin{bmatrix} 1 & 0 \\ 1 & -2 \end{bmatrix}$$

both ways described above (*i.e.*, find P and Q and then calculate the R 's, and then do a partial fraction expansion of $(sI - A)^{-1}$ to find the R 's).

3. Consider the discrete-time system $x(t+1) = Ax(t)$, where $x(t) \in \mathbf{R}^n$.
- (a) Find $x(t)$ in terms of $x(0)$.
- (b) Suppose that $\det(zI - A) = z^n$. What are the eigenvalues of A ? What (if anything) can you say about $x(k)$ for $k < n$ and $k \geq n$, without knowing $x(0)$?
4. Consider the linear dynamical system $\dot{x} = Ax$ where $A \in \mathbf{R}^{n \times n}$ is diagonalizable with eigenvalues λ_i , eigenvectors v_i , and left eigenvectors w_i for $i = 1, \dots, n$. Assume that $\lambda_1 > 0$ and $\Re \lambda_i < 0$ for $i = 2, \dots, n$. Describe the trajectories qualitatively. Specifically, what happens to $x(t)$ as $t \rightarrow \infty$? Give the answer geometrically, in terms of $x(0)$.

5. *Stability of a periodic system.* Consider the linear dynamical system $\dot{x} = A(t)x$ where

$$A(t) = \begin{cases} A_1 & 2n \leq t < 2n + 1, \quad n = 0, 1, 2, \dots \\ A_2 & 2n + 1 \leq t < 2n + 2, \quad n = 0, 1, 2, \dots \end{cases}$$

In other words, $A(t)$ switches between the two values A_1 and A_2 every second. We say that this (time-varying) linear dynamical system is stable if every trajectory converges to zero, *i.e.*, we have $x(t) \rightarrow 0$ as $t \rightarrow \infty$ for any $x(0)$.

Find the conditions on A_1 and A_2 under which the periodic system is stable. Your conditions should be as explicit as possible.

6. *Rate of a Markov code.* Consider the Markov language described in exercise 2-3, with five symbols 1, 2, 3, 4, 5, and the following symbol transition rules:

- 1 must be followed by 2 or 3
- 2 must be followed by 2 or 5
- 3 must be followed by 1
- 4 must be followed by 4 or 2 or 5
- 5 must be followed by 1 or 3

(a) *The rate of the code.* Let K_N denote the number of allowed sequences of length N . The number

$$R = \lim_{N \rightarrow \infty} \frac{\log_2 K_N}{N}$$

(if it exists) is called the *rate* of the code, in bits per symbol. Find the rate of this code. Compare it to the rate of the code which consists of all sequences from an alphabet of 5 symbols (*i.e.*, with no restrictions on which symbols can follow which symbols).

(b) *Asymptotic fraction of sequences with a given starting or ending symbol.* Let $F_{N,i}$ denote the number of allowed sequences of length N that start with symbol i , and let $G_{N,i}$ denote the number of allowed sequences of length N that end with symbol i . Thus, we have

$$F_{N,1} + \dots + F_{N,5} = G_{N,1} + \dots + G_{N,5} = K_N.$$

Find the asymptotic fractions

$$f_i = \lim_{N \rightarrow \infty} F_{N,i}/K_N, \quad g_i = \lim_{N \rightarrow \infty} G_{N,i}/K_N.$$

We won't give full credit for answers obtained by simple simulation or relatively mindless computation; we want to see (and understand) your method.

7. *Squareroot and logarithm of a (diagonalizable) matrix.* Suppose that $A \in \mathbf{R}^{n \times n}$ is diagonalizable. Therefore, an invertible matrix $T \in \mathbf{C}^{n \times n}$ and diagonal matrix $\Lambda \in \mathbf{C}^{n \times n}$ exist such that $A = T\Lambda T^{-1}$. Let $\Lambda = \mathbf{diag}(\lambda_1, \dots, \lambda_n)$.

- (a) We say $B \in \mathbf{R}^{n \times n}$ is a squareroot of A if $B^2 = A$. Let μ_i satisfy $\mu_i^2 = \lambda_i$. Show that $B = T \mathbf{diag}(\mu_1, \dots, \mu_n) T^{-1}$ is a squareroot of A . A squareroot is sometimes denoted $A^{1/2}$ (but note that there are in general many squareroots of a matrix).
When λ_i are real and nonnegative, it is conventional to take $\mu_i = \sqrt{\lambda_i}$ (*i.e.*, the nonnegative squareroot), so in this case $A^{1/2}$ is unambiguous.
- (b) We say B is a logarithm of A if $e^B = A$, and we write $B = \log A$. Following the idea of part a, find an expression for a logarithm of A (which you can assume is invertible). Is the logarithm unique? What if we insist on B being real?
8. Suppose $\dot{x} = Ax$ with $A \in \mathbf{R}^{n \times n}$. Two one-second experiments are performed. In the first, $x(0) = [1 \ 1]^T$ and $x(1) = [4 \ -2]^T$. In the second, $x(0) = [1 \ 2]^T$ and $x(1) = [5 \ -2]^T$.
- (a) Find $x(1)$ and $x(2)$, given $x(0) = [3 \ -1]^T$.
- (b) Find A , by first computing the matrix exponential.
- (c) Either find $x(1.5)$ or explain why you cannot ($x(0) = [3 \ -1]^T$).
- (d) More generally, for $\dot{x} = Ax$ with $A \in \mathbf{R}^{n \times n}$, describe a procedure for finding A using experiments with different initial values. What conditions must be satisfied for your procedure to work?
9. *Affine dynamical systems.* A function $f : \mathbf{R}^n \rightarrow \mathbf{R}^m$ is called *affine* if it is a linear function plus a constant, *i.e.*, of the form $f(x) = Ax + b$. Affine functions are more general than linear functions, which result when $b = 0$. We can generalize linear dynamical systems to *affine dynamical systems*, which have the form

$$\dot{x} = Ax + Bu + f, \quad y = Cx + Du + g.$$

Fortunately we don't need a whole new theory for (or course on) affine systems; a simple shift of coordinates converts it to a linear dynamical system. Assuming A is invertible, define $\tilde{x} = x + A^{-1}f$ and $\tilde{y} = y - g + CA^{-1}f$. Show that \tilde{x} , u , and \tilde{y} are the state, input, and output of a linear dynamical system.

10. *Analysis of a power control algorithm.* In this problem we consider again the power control method described in homework problem 1-1. Please refer to this problem for the setup and background. In that problem, you expressed the power control method as a discrete-time linear dynamical system, and simulated it for a specific set of parameters, with several values of initial power levels, and two target SINRs. You found that for the target SINR value $\gamma = 3$, the powers converged to values for which each SINR exceeded γ , no matter what the initial power was, whereas for the larger target SINR value $\gamma = 5$, the powers appeared to diverge, and the SINRs did not appear to converge. You are going to analyze this, now that you know a lot more about linear systems.
- (a) *Explain the simulations.* Explain your simulation results from the problem 1(b) for the given values of G , α , σ , and the two SINR threshold levels $\gamma = 3$ and $\gamma = 5$.

- (b) *Critical SINR threshold level.* Let us consider fixed values of G , α , and σ . It turns out that the power control algorithm works provided the SINR threshold γ is less than some critical value γ_{crit} (which might depend on G , α , σ), and doesn't work for $\gamma > \gamma_{\text{crit}}$. ('Works' means that no matter what the initial powers are, they converge to values for which each SINR exceeds γ .)

Find an expression for γ_{crit} in terms of $G \in \mathbf{R}^{n \times n}$, α , and σ . Give the simplest expression you can. Of course you must explain how you came up with your expression.