Inference, Control & Driving in Natural Systems - Assignments

Tom McGrath

January 13, 2014

1 Rejection Sampling in higher-dimensional problems

Rejection sampling can be done when we have a distribution $P(x) = P*(x)/Z$ which it's not possible to sample from directly, and another, simpler density $Q(x)$ (called the proposal density) which can be sampled from where there exists a constant c such that:

$$cQ^*(x) > P^*(x)\ \forall x \tag{1}$$

We sample first from $Q(x)$, then use this x to sample once from the uniform distribution of $[0, cQ^*(x)]$, and call the sample value u. If $u > P^*(x)$ then we accept u and add it to our sample set, otherwise we reject it. This procedure samples from $P(x)$. The fewer x values that are rejected, the faster rejection sampling will work - this implies that $cQ^*(x)$ needs to closely approximate $P^*(x)$ to work well, and the more space between the two distributions causes slower sampling.

In general, higher dimensionality leads to more difference between distributions, causing rejection sampling to work slower in higher dimensions. We can illustrate this using the example of multidimensional Gaussians given in [1]. Sample from an N-dimensional Gaussian distribution with standard deviation σ_Q to approximate one with s.d. σ_P. $Q(\mathbf{x})$ has density at the origin:

$$\rho(\mathbf{x}) = \frac{1}{(2\pi\sigma_Q^2)^N/2} \tag{2}$$

To fulfil the inequality needed for rejection sampling we need to have:

$$c = \frac{(2\pi\sigma_Q^2)^N/2}{(2\pi\sigma_P^2)^N/2} = \exp\left(N \ln \frac{\sigma_Q}{\sigma P}\right) \tag{3}$$

The acceptance rate r is:

$$r = \frac{\int P(x)}{\int cQ(x)} = \frac{1}{c} \tag{4}$$

as $P(x)$ and $Q(x)$ are both normalised. In the N-dimensional Gaussian example we thus have:

$$r = \exp\left(-N \ln \frac{\sigma_Q}{\sigma_P}\right) \tag{5}$$

So acceptance rate decreases exponentially with N. According to [1] this is true in general.

2 Proof of Landauer's principle

Landauer's principle, first stated in [2], places a lower bound on the entropy generation of any logically irreversible computation. Computation must be performed on information stored in the computing system, and the result of the computation will depend on the information stored. If we consider a binary system of N bits, then the system has 2^N degrees of freedom. A reversible computation maps these 2^N states to the same set of 2^N states bijectively (otherwise it isn't reversible). This constrains the possible operations to those which have equal numbers of zeros and ones - the only computations that satisfy this requirement are the identity, XOR and their negations. These aren't sufficient for computation as we typically consider it, and we can't reconstruct other truth functions by combining them.

It is possible to 'embed' an irreversible truth function in a reversible truth function with a larger state space, where the extra space is used to store information about the truth function in such a way as to make it reversible. But there are 2 problems with this - firstly, the amount of storage space required grows continuously as the computation goes on, meaning every program must terminate or eventually run out of memory. Secondly, this information must be stored somewhere, and this storage area must be reset before computation occurs. Instead of avoiding irreversibility, we've just changed the time at which it occurs.

Therefore, we have to accept some irreversibility in computation. If we take a system of bits in thermal equilibrium, and want to specify a particular bit as one (or, equivalently, zero), then the number of states available is half that before the state of this bit was specified. This creates a change in entropy of $klog2$. In a closed system entropy cannot decrease, so this entropy must be transferred elsewhere, in non-information-carrying degrees of freedom (i.e. heat). This gives a minimum heating effect of $k_B T log2$ per bit, which is commonly referred to as the Landauer limit.

3 Glauber dynamics

Glauber dynamics is a Markov Chain Monte Carlo method of simulating the dynamics of a system. A system is specified (often this is spins on the Ising lattice), and an energy is defined over the possible states. The dynamics of the system are the probability of changing from one state to another, and the choice of dynamics called Glauber dynamics gives the probability of changing from a state of energy E_{old} to E_{new} as:

$$p = \frac{e^{-E_{new}/k_B T}}{e^{-E_{old}/k_B T} + e^{-E_{new}/k_B T}} \tag{6}$$

For a system of spins simulating a physical system, the Monte Carlo timestep is so short that it would take approximately 10^{13} steps per lattice site to simulate dynamics of one second [3]. This is too slow for many physical systems of interest, so more advanced algorithms are needed to get results in a reasonable time.

4 PID control of an inverted pendulum

An inverted pendulum is a pendulum which is stabilised at the 'top' of its range of motion. This is an unstable local equilibrium and any disturbance not counterbalanced by a control law will rapidly lead to the pendulum swinging away from this point. The system considered in [4] is the pendulum linearised above its upper position, with some unknown but constant disturbance e and control $u(t)$:

$$\ddot{\psi}(t) - \psi(t) = u(t) + e \tag{7}$$

We want to control this system such that $\psi(t) \to 0$ as $t \to \infty$. In the absence of disturbances ($e = 0$), we can stabilise this system using proportional-derivative control:

$$u(t) = -\alpha\psi(t) - \beta\dot{\psi}(t) \tag{8}$$

A proportional-derivative control leads to the dynamical system:

$$\ddot{\psi}(t) + \beta\dot{\psi}(t) + (\alpha - 1)\psi(t) = e \tag{9}$$

This has no solution where $\psi(t) \to 0$ as $t \to \infty$, as we can see by solving the differential equation. The characteristic polynomial of the homogeneous system (without the disturbance) is:

$$z^2 + \beta z + (\alpha - 1) = 0 \tag{10}$$

Solving for z:

$$z_\pm = \frac{-\beta \pm \sqrt{\beta^2 - 4(\alpha - 1)}}{2} \tag{11}$$

This gives solutions which decay in time if $\alpha > 1$ and $\beta > 0$. However, considering the particular solution shows that instead of $\psi(t) \to 0$ as $t \to \infty$ we obtain $\psi(t) \to \frac{e}{\alpha - 1}$ as $t \to \infty$. Qualitatively, the system will be stabilised, but in the wrong place; the constant disturbance e pushes it away. Adding integral control lets us detect such 'systematic' errors, which are eliminated by the PID control law:

$$u(t) = -\alpha\psi(t) - \beta\dot{\psi}(t) - \mu\int_0^t \psi(\tau)d\tau \tag{12}$$

5 Proof of matrix requirements for controllability (Zabczyc)

The most intuitive presentation of this material I've found is a series of online lecture notes at [5]. A system of the form

$$\dot{x} = Ax + Bu \tag{13}$$

is controllable on $[0, T]$ if, for η in $\mathbb{R}^n$:

$$\eta^T \int_0^T e^{A(T-\tau)} Bu(\tau d\tau) = 0 \quad \forall u(\cdot) \in L^1(0,T) \tag{14}$$

This implies $\eta = 0$ if the system is controllable - controllability implies that the set of reachable states $\bar{x} \in \mathbb{R}^n$ for any choice of control u is the whole of $\mathbb{R}^n$. From this, we need to have

$$\eta^T e^{A(T-\tau)} B = 0 \quad \forall \tau \in [0,T] \tag{15}$$

Otherwise we can set $u(\tau) = B^T e^{A^T(T-\tau)} \eta \neq 0$ and substitute this into the equation above to obtain:

$$\int_0^T u^T(\tau) u(\tau) d\tau = \int_0^T ||u(\tau)||^2 d\tau > 0 \tag{16}$$

in contradiction of the controllability condition above. Now take the j-th derivative with respect to τ:

$$\frac{d^j}{d\tau^j} = (-1)^j \eta^T A^j e^{A(T-\tau)} B = 0 \tag{17}$$

At $T = \tau$ we have:

$$(-1)^j \eta^T A^j e^{A(T-\tau)} B = (-1)^j \eta^T A^j B = 0 \tag{18}$$

Therefore (as we can take the derivative $n-1$ times):

$$\eta^T [B \; AB \; ... \; A^{n-1}B] = 0 \tag{19}$$

As $\eta = 0$ and the matrix above is n by n:

$$rank([B \; AB \; ... \; A^{n-1}B]) = n \tag{20}$$

and the matrix is invertible.

6 Ott, Grebogi & Yorke algorithm for stabilising chaotic dynamics

Chaotic systems are, by definition, nonlinear, so control methods for linear systems aren't generally applicable. The Ott, Grebogi & Yorke algorithm [6] for stabilising chaotic systems relies on the ergodicity of the system on its attractor, and the existence of a dense set of unstable periodic orbits (including fixed points).

The algorithm works as follows: first pick an unstable fixed point to control the system towards. Because the system is ergodic over the attractor set, it will eventually pass arbitrarily close to this fixed point. Once the system is close enough that linearisation around the fixed point is a suitable approximation, apply a control law to the linearised system to control it to the fixed point and keep it there.

The case of a discretised system is given in [7]. If the system has dynamics $x_{n+1} = f(\lambda, x_n)$ for some parameter λ then the linearised dynamics around fixed point x^* are:

$$x_{n+1} - x^* \simeq \frac{\partial f}{\partial x}(x_n - x^*) + \frac{\partial f}{\partial \lambda}\Delta\lambda_n \tag{21}$$

To control the system to x^*, alter λ such that $x_{n+1} = x^*$ by setting:

$$\Delta\lambda_n = -\frac{\partial f(x_n - x^*)/\partial x}{\partial f/\partial \lambda} \tag{22}$$

The linearised system is not a perfect approximation, so in general this will not be successful in one step. However, if the linearised control law is applied when the system is 'close enough' to x^* then it will quickly converge.

7 Proof of Bode's integral formula

Bode's integral formula (also known as the 'waterbed effect') states that for the loop transfer function $L(s)$ of a feedback system which goes to zero faster than $1/s$ as $s \to \infty$ with poles p_k in the right-half plane:

$$\int_0^\infty \log|S(i\omega)|d\omega = \int_0^\infty \log\frac{1}{|1 + L(i\omega)|}d\omega = \pi\sum_k p_k \tag{23}$$

We can prove this by considering integration around the contour below (the image and proof are from [8]:

We can divide the integral into 3 parts:

$$\begin{aligned}\int_\Gamma \log(S(s))\,ds &= \int_{-iR}^{iR}\log(S(s))\,ds + \int_R \log(S(s))\,ds + \sum_k\int_\gamma \log(S(s))\,ds && (24)\\ &= I_1 + I_2 + I_3 && (25)\end{aligned}$$

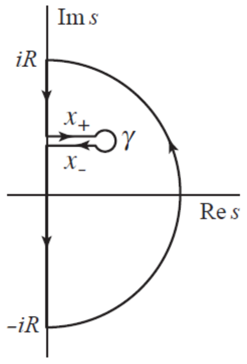

Figure 1: Contour integral for Bode's integral theorem with a single pole. From [8]

Now we can calculate term by term. I_1 can be written:

$$I_1 = -i \int_{iR}^{iR} \log(S(i\omega))\, d\omega = -2i \int_0^{iR} \log(|S(i\omega|)|)\, d\omega \tag{26}$$

By writing I_2 in terms of $L(s)$ and using the approximation $\log(1 + x) \approx x$ for small x:

$$I_2 = \int_R los(S(s))\, ds = -\int_R \log(1 + L(s))\, ds \approx -\int_R L(s)\, ds \tag{27}$$

This tends to zero faster than $1/s$, so $I_2 \to 0$ as $R \to \infty$. Finally, we consider I_3 in terms of the three parts of the 'cut-out': X_+, X_- and γ:

$$I_3 = \int_{X_+} log(S(s))\, ds + \int_\gamma log(S(s))\, ds + \int_{X_-} log(S(s))\, ds \tag{28}$$

The integral around γ goes to zero, leaving. Evaluating the X_+ and X_- terms we get:

$$I_3 = 2\pi i \sum_k \Re(p_k) \tag{29}$$

This gives:

$$-2i \int_0^R \log|S(i\omega)|\, d\omega + i \sum_k 2\pi \Re(p_k) \tag{30}$$

Writing $S(i\omega)$ in terms of $L(i\omega)$ and dividing through by $2i$ gives the result.

8 Derive optimal control and cost-to-go of LQG control

Following the presentation given in [9]: Optimal control of a Linear Quadratic Gaussian system gives a closed-form control law. With the control problem:

$$\textit{dynamics: } d\mathbf{x} = (A\mathbf{x} + B\mathbf{u})dt + Fd\mathbf{w} \tag{31}$$

$$\textit{cost rate: } l(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{u}^T R\mathbf{u} + \frac{1}{2}\mathbf{x}^T Q\mathbf{x} \tag{32}$$

$$\textit{final cost: } h(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q^f \mathbf{x} \tag{33}$$

We seek a control law which satisfies the stochastic Hamilton-Jacobi-Bellman equations for optimal control law $\pi(\mathbf{x}, t)$ and optimal cost-to-go $v(\mathbf{x}, t)$:

$$\pi(\mathbf{x}, t) = \arg\min_{\mathbf{x}\in\mathcal{U}} \Big(l(\mathbf{x}, \mathbf{u}, t) + \mathbf{f}(\mathbf{x}, \mathbf{u})^T v_{\mathbf{x}}(\mathbf{x}, t) + \frac{1}{2} tr(S(\mathbf{x}, \mathbf{u}) v_{\mathbf{xx}}(\mathbf{x}, t) \Big) \tag{34}$$

$$-v_t(\mathbf{x}, t) = \min_{\mathbf{x}\in\mathcal{U}} \Big(l(\mathbf{x}, \mathbf{u}, t) + \mathbf{f}(\mathbf{x}, \mathbf{u})^T v_{\mathbf{x}}(\mathbf{x}, t) + \frac{1}{2} tr(S(\mathbf{x}, \mathbf{u}) v_{\mathbf{xx}}(\mathbf{x}, t) \Big) \tag{35}$$

Making the Ansatz for the optimal cost-to-go with some unknown $a(t)$ we hope to obtain later:

$$v(\mathbf{x}, t) = \frac{1}{2}\mathbf{x}^T V(t)\mathbf{x} + a(t) \tag{36}$$

and inserting it into the HJB equation we recover the optimal control:

$$\mathbf{u} = -R^{-1}B^T V(t)\mathbf{x} \tag{37}$$

This satisfies the $min_{\mathbf{u}}$ operator in the HJB equation, in which we can now equate the $\mathbf{x}$-dependent terms and obtain a formula for $\dot{V}(t)$ and $\dot{a}(t)$ to give the optimal cost-to-go.

$$-\dot{V(t)} = Q + A^T V(t) + V(t)A - V(t)BR^{-1}B^T V(t) \tag{38}$$

This is called a continuous-time Ricatti equation.

$$-\dot{a}(t) = \frac{1}{2} trace(SV(t)) \tag{39}$$

Will give the optimal cost-to-go $v(\mathbf{x}, t)$ when integrated to give $a(t)$.

9 The Simplex Algorithm

Dantzig's Simplex algorithm is a linear programming algorithm for optimising a function of multiple variables given constraints on those variables. For example (from the webpage given in the question) maximising:

$$\max 5x_1 + 4x_2 + 3x_3 \tag{40}$$

Given the constraints:

$$2x_1 + 3x_2 + x_3 \leq 5 \tag{41}$$
$$4x_1 + x_2 + 2x_3 \leq 11 \tag{42}$$
$$3x_1 + 4x_2 + 2x_3 \leq 8 \tag{43}$$
$$x_1, x_2, x_3 \geq 0 \tag{44}$$

We contain the objective within a variable z:

$$z = 5x_1 + 4x_2 + 3x_3 \tag{45}$$

The constraints to the x_i are encoded in linear combinations of the initial variables, which we call 'slack variables':

$$x_4 = 5 - 2x_1 - 3x_2 - x_3 \tag{46}$$
$$x_5 = 11 - 4x_1 - x_2 - 2x_3 \tag{47}$$
$$x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \tag{48}$$

Together with the equation for z this is called a dictionary for the linear optimisation problem. We can represent all this information in an augmented matrix called a 'tableau', written:

$$\begin{bmatrix} 0 & A & I & b \\ -1 & c^T & 0 & 0 \end{bmatrix} \tag{49}$$

where A is the matrix of coefficients of constraints (the LHS of the constraint equations), b is the values on the RHS of the constraint equations, c the vector of coefficients of the objective variable and I the identity matrix. We then choose a column to eliminate (the 'pivot' column) and perform Gaussian elimination on the augmented matrix using this column. Continue pivoting until all the elements of c^T are less than zero - this is an optimal solution as increasing any of the variables will decrease the objective variable. It is also possible to compute a matrix ('Gaussian pivot matrix') for each step such that multiplying the simplex tableau by this matrix will perform the desired pivot.

10 Molecular Democracy: who shares the controls?

From [10] Summation theorem:

Assume that the expression for the rate of a reaction is a function of (at most) the substrates S_i, inhibitors I_i, cofactors Co_i, rate constants k_i and equilibrium constants K_i. Then the sum of the sensitivity coefficients $\mathbf{Z}_i$ is unity:

$$\sum_{i=1}^{n} \mathbf{Z}_{E_i}^{F_j} = 1 \tag{50}$$

We can show this by considering the situation at steady-state: the intermediate metabolite concentrations are constant. Increasing all enzymes by some small α doesn't change metabolite concentrations:

$$\frac{\delta E_i}{E_i} = \alpha \implies S_i = constant \tag{51}$$

It does, however, change fluxes:

$$\frac{\delta F_j}{F_j} = \alpha \tag{52}$$

Now taking partial derivatives:

$$\frac{dF_j}{F_j} = \sum_{i=1}^{n} \left(\frac{\delta F_j}{F_j} \bigg/ \frac{\delta E_i}{E_i} \right) \frac{dE_i}{E_i} \tag{53}$$

Recognising the partial derivatives as the sensitivity coefficients $\mathbf{Z_i}$ and taking the limit:

$$\alpha = \alpha \sum_{i=1}^{n} \mathbf{Z}_i \implies \sum_{i=1}^{n} \mathbf{Z}_i = 1 \tag{54}$$

Connectivity theorem:

Define the elasticity coefficient:

$$\frac{\delta v_i}{v_i} \bigg/ \frac{\delta S_i}{S_i} \to \frac{\partial \ln v_i}{\partial \ln S_i} = \varepsilon_{S_i}^{v_i} \tag{55}$$

The connectivity theorem states that:

$$\frac{\mathbf{Z}_i}{\mathbf{Z}_j} = -\frac{\varepsilon^j}{\varepsilon^i} \tag{56}$$

We can see this by substituting $dv/v = dE/E$ into the definition of the elasticity coefficient for enzymes i and j which share a common linking metabolite and inserting this into the expression for the system flux:

$$\frac{dF}{F} = 0 = \mathbf{Z}_i \frac{dE_i}{E_i} + \mathbf{Z}_j \frac{dE_j}{E_j} \tag{57}$$

This gives:

$$\mathbf{Z}_i \varepsilon^i + \mathbf{Z}_j \varepsilon^j = 0 \tag{58}$$

Dividing through gives the result.

11 Derivation of the Michaelis-Menten expression

This derivation follows that given on the webpage in the question. Consider an enzyme E that catalyses a reaction from substrate S to product P:

$$E + S \underset{\mathrm{k_{-1}}}{\overset{\mathrm{k_1}}{\rightleftarrows}} ES \overset{\mathrm{k_{cat}}}{\rightarrow} E + P \tag{59}$$

Assume the reaction has first-order kinetics, so rate v_0 has:

$$v_0 = \frac{d[P]}{dt} = k_{cat}[ES] \tag{60}$$

If the reaction is at steady state, then the rate of formation of the intermediate ES (LHS below) equals its rate of consumption (RHS below):

$$k_{-1}[ES] + k_{cat}[ES] = k_1[E][S] \tag{61}$$

Now we find:

$$\frac{k_{-1} + k_{cat}}{k_1} = \frac{[E][S]}{[ES]} \tag{62}$$

Grouping together the rate constants into one constant $K_m = (k_{-1} + k_{cat}/k_1)$ and writing $[E] = [E]_{total} - [ES]$ we obtain:

$$K_m = \frac{([E]_{total} - [ES])[S]}{[ES]} \tag{63}$$

Rearranging:

$$[ES] = \frac{[E]_{total}[S]}{K_m + [S]} \tag{64}$$

The rate of reaction has a maximum velocity v_{max} when all enzymes are saturated, i.e. $[ES] = [E]_{total}$. This means $v_{max} = k_{cat}[E]_{total}$. Substituting this into our initial equation:

$$v_0 = \frac{v_{max}[S]}{K_m + [S]} \tag{65}$$

This is the Michaelis-Menten equation. The Michaelis constant K_m is the substrate concentration where the rate of reaction is half v_{max}. The reaction rate will asymptotically approach v_{max} with increasing substrate saturation - the reaction rate is a rectangular hyperbola.

References

[1] MacKay, D. J. *Information theory, inference and learning algorithms.* Cambridge university press, (2003).

[2] Landauer, R. *IBM Journal of Research and Development* **44**(1.2), 261–269 (2000).

[3] Novotny, M. A. *arXiv preprint cond-mat/0109182* (2001).

[4] Sontag, E. D. *Mathematical control theory: deterministic finite dimensional systems*, volume 6. Springer, (1998).

[5] Xia, H. (1995).

[6] Ott, E., Grebogi, C., and Yorke, J. A. *Controlling chaos: theoretical and practical methods in non-linear dynamics* , 77 (1996).

[7] Bechhoefer, J. *Reviews of Modern Physics* **77**(3), 783 (2005).

[8] Aström, K. J. and Murray, R. M. *Feedback systems: an introduction for scientists and engineers.* Princeton university press, (2010).

[9] Todorov, E. *Bayesian brain: probabilistic approaches to neural coding* , 269–298 (2006).

[10] Kacser, H., Burns, J., et al. *Biochem. Soc. Trans* **7**(1), 149–1 (1979).