5. Linear models

- linear and affine functions
- Taylor approximation
- regression model
- linear equations
- linear dynamical systems

Linear functions

- $f : \mathbb{R}^n \to \mathbb{R}^m$ means f is a function mapping n-vectors to m-vectors
- value is an m-vector $f(x) = (f_1(x), \ldots, f_m(x))$
- example: $f(x) = (x_1^2, x_2 - x_1, x_2)$ is $f : \mathbb{R}^2 \to \mathbb{R}^3$

Linear functions: f is linear if it satisfies the *superposition* property

$$f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$$

for all numbers α, β, and all n-vectors x, y

Extension: if f is linear, then

$$f\left(\alpha_1 u_1 + \alpha_2 u_2 + \cdots + \alpha_m u_m\right) = \alpha_1 f\left(u_1\right) + \alpha_2 f\left(u_2\right) + \cdots + \alpha_m f\left(u_m\right)$$

for all n-vectors $u_1, \ldots, u_m$ and all scalars $\alpha_1, \ldots, \alpha_m$

Matrix-vector product function

define a function $f : \mathbb{R}^n \to \mathbb{R}^m$ as $f(x) = Ax$ for fixed $A \in \mathbb{R}^{m\times n}$

- any function of this type is linear: $A(\alpha x + \beta y) = \alpha(Ax) + \beta(Ay)$
- every linear function f can be written as $f(x) = Ax$:

$$\begin{aligned} f(x) &= f\left(x_1e_1 + x_2e_2 + \cdots + x_ne_n\right) \\ &= x_1f(e_1) + x_2f(e_2) + \cdots + x_nf(e_n) \\ &= [f(e_1)\ f(e_2)\ \cdots\ f(e_n)] \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = Ax \end{aligned}$$

  where $A = [f(e_1)\ \ f(e_2)\ \ \cdots\ \ f(e_n)]$ and $f(e_i)$ is an m-vector

- for $f : \mathbb{R}^n \to \mathbb{R}$, we get inner product function $f(x) = a^Tx$
- for any linear function f there is only one A for which $f(x) = Ax$ for all x

Examples ($f : \mathbb{R}^3 \to \mathbb{R}^3$)

Linear

- f reverses the order of the components of x is linear

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

- f scales x_1 by a given number d_1, x_2 by d_2, x_3 by d_3 is linear

$$A = \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix}$$

Nonlinear

- f sorts the components of x in decreasing order: not linear
- f replaces each x_i by its absolute value $|x_i|$: not linear

Composition of linear functions

- A is an $m \times p$ matrix
- B is $p \times n$
- define linear functions $f : \mathbb{R}^p \longrightarrow \mathbb{R}^m$ and $g : \mathbb{R}^n \longrightarrow \mathbb{R}^p$ as

$$f(u) = Au, \quad g(v) = Bv$$

- *composition* of f and g is $h : \mathbb{R}^n \longrightarrow \mathbb{R}^m$

$$h(x) = f(g(x)) = A(Bx) = (AB)x$$

- composition of linear functions is linear
- associated matrix is product of matrices of the functions

Example: Second difference matrix

- D_n is $(n-1) \times n$ difference matrix:

$$D_n x = (x_2 - x_1, x_3 - x_2, \ldots, x_n - x_{n-1})$$

- D_{n-1} is $(n-2) \times (n-1)$ difference matrix:

$$D_n y = (y_2 - y_1, y_3 - y_2, \ldots, y_{n-1} - y_{n-2})$$

- $\Delta = D_{n-1} D_n$ is $(n-2) \times n$ is called *second difference* matrix:

$$\Delta x = (x_1 - 2x_2 + x_3, x_2 - 2x_3 + x_4, \ldots, x_{n-2} - 2x_{n-1} + x_n)$$

- for $n = 5, \Delta = D_{n-1} D_n$ is

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

Affine function

a function $f : \mathbb{R}^n \to \mathbb{R}^m$ is *affine* if it satisfies

$$f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$$

for all n-vectors x, y and all scalars α, β with $\alpha + \beta = 1$

Extension: if f is affine, then

$$f\left(\alpha_1 u_1 + \alpha_2 u_2 + \cdots + \alpha_m u_m\right) = \alpha_1 f\left(u_1\right) + \alpha_2 f\left(u_2\right) + \cdots + \alpha_m f\left(u_m\right)$$

for all n-vectors $u_1, \ldots, u_m$ and all scalars $\alpha_1, \ldots, \alpha_m$ with

$$\alpha_1 + \alpha_2 + \cdots + \alpha_m = 1$$

Affine functions and matrix-vector product

$f : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ is affine, if and only if it can be expressed as

$$f(x) = Ax + b$$

for some $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$

- to see it is affine, let $\alpha + \beta = 1$ then

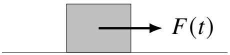
$$A(\alpha x + \beta y) + b = \alpha(Ax + b) + \beta(Ay + b)$$

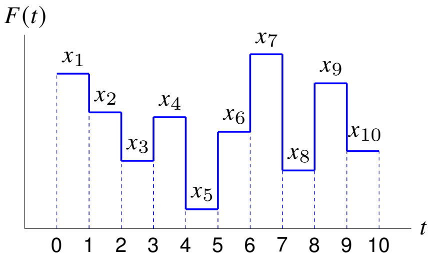
- using the definition, we can show

$$A = [f(e_1) - f(0) \;\; f(e_2) - f(0) \;\; \cdots \;\; f(e_n) - f(0)], \quad b = f(0)$$

- for $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ the above becomes $f(x) = a^T x + b$

Example: Motion of a mass





- a unit mass with zero initial position and velocity
- we apply piecewise-constant force $F(t)$ during interval $[0, 10)$:

$$F(t) = x_j \quad \text{for } t \in [j-1, j), \quad j = 1, \ldots, 10$$

- define $f(x)$ as position at $t = 10$, $g(x)$ as velocity at $t = 10$

find f and g and determine whether they are linear or affine in x?

Solution

- from Newton's law $p''(t) = F(t)$ where $p(t)$ is the position at time t
- integrate to get final velocity and position

$$\begin{aligned} g(x) = p'(10) &= \int_0^{10} F(t)dt \\ &= x_1 + x_2 + \cdots + x_{10} \\ f(x) = p(10) &= \int_0^{10} p'(t)dt \\ &= \frac{19}{2}x_1 + \frac{17}{2}x_2 + \frac{15}{2}x_3 + \cdots + \frac{1}{2}x_{10} \end{aligned}$$

- the two functions are linear: $f(x) = a^T x$ and $g(x) = b^T x$ with

$$a = \left(\frac{19}{2}, \frac{17}{2}, \ldots, \frac{3}{2}, \frac{1}{2}\right), \quad b = (1, 1, \ldots, 1)$$

Outline

- linear and affine functions
- **Taylor approximation**
- regression model
- linear equations
- linear dynamical systems

First-order Taylor (affine) approximation

first-order *Taylor approximation* of $f : \mathbb{R}^n \to \mathbb{R}$, near point z:

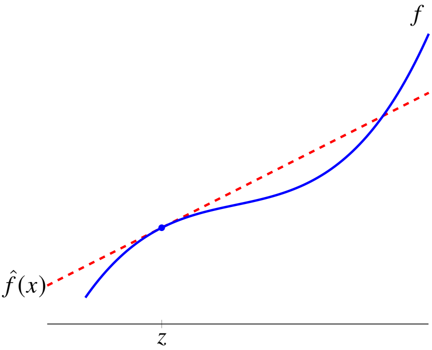
$$\begin{aligned}\hat{f}(x) &= f(z) + \frac{\partial f}{\partial x_1}(z)\,(x_1 - z_1) + \cdots + \frac{\partial f}{\partial x_n}(z)\,(x_n - z_n) \\ &= f(z) + \nabla f(z)^T (x - z)\end{aligned}$$

- n-vector $\nabla f(z)$ is the *gradient* of f at z,

$$\nabla f(z) = \left(\frac{\partial f}{\partial x_1}(z), \ldots, \frac{\partial f}{\partial x_n}(z) \right)$$

- $\hat{f}(x)$ is very close to $f(x)$ when x_i are all near z_i
- sometimes written $\hat{f}(x; z)$, to indicate that z where the approximation appear
- $\hat{f}$ is an affine function of x
- often called *linear approximation* of f near z, even though it is in general affine

Example with one variable



$$\hat{f}(x) = f(z) + f'(z)(x - z)$$

Example with two variables

$$f(x_1, x_2) = x_1 - 3x_2 + e^{2x_1+x_2-1}$$

- gradient:

$$\nabla f(x) = \begin{bmatrix} 1 + 2e^{2x_1+x_2-1} \\ -3 + e^{2x_1+x_2-1} \end{bmatrix}$$

- Taylor approximation around $z = 0$:

$$\begin{aligned} \hat{f}(x) &= f(0) + \nabla f(0)^T (x - 0) \\ &= e^{-1} + (1 + 2e^{-1})x_1 + (-3 + e^{-1})x_2 \end{aligned}$$

Taylor approximation for vector-valued functions

first-order Taylor approximation of differentiable $f : \mathbb{R}^n \to \mathbb{R}^m$ around z:

$$\hat{f}_i(x) = f_i(z) + \frac{\partial f_i}{\partial x_1}(z)\,(x_1 - z_1) + \cdots + \frac{\partial f_i}{\partial x_n}(z)\,(x_n - z_n)\,, \quad i = 1, \ldots, m$$

in matrix-vector notation: $\hat{f}(x) = f(z) + Df(z)(x - z)$ where

$$Df(z) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(z) & \frac{\partial f_1}{\partial x_2}(z) & \cdots & \frac{\partial f_1}{\partial x_n}(z) \\ \frac{\partial f_2}{\partial x_1}(z) & \frac{\partial f_2}{\partial x_2}(z) & \cdots & \frac{\partial f_2}{\partial x_n}(z) \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_m}{\partial x_1}(z) & \frac{\partial f_m}{\partial x_2}(z) & \cdots & \frac{\partial f_m}{\partial x_n}(z) \end{bmatrix} = \begin{bmatrix} \nabla f_1(z)^T \\ \nabla f_2(z)^T \\ \vdots \\ \nabla f_m(z)^T \end{bmatrix}$$

- $Df(z)$ is called the *derivative* or *Jacobian* matrix of f at z
- $\hat{f}$ is a local affine approximation of f around z

Example

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} = \begin{bmatrix} e^{2x_1+x_2} - x_1 \\ x_1^2 - x_2 \end{bmatrix}$$

- derivative matrix

$$Df(x) = \begin{bmatrix} 2e^{2x_1+x_2} - 1 & e^{2x_1+x_2} \\ 2x_1 & -1 \end{bmatrix}$$

- first order approximation of f around $z = 0$:

$$\hat{f}(x) = \begin{bmatrix} \hat{f}_1(x) \\ \hat{f}_2(x) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Outline

- linear and affine functions
- Taylor approximation
- **regression model**
- linear equations
- linear dynamical systems

Regression model

a *regression* model is the affine function:

$$\hat{y} = x^T\beta + v = \beta_1 x_1 + \cdots + \beta_n x_n + v$$

- $\hat{y}$ is *prediction* of true value y called the *dependent variable*, *outcome*, or *label*
- x is *regressor* or *feature* vector (entries called regressors)
- β is *weight* or *coefficient* vector (β_i are model parameters)
- v is *offset* parameter or *intercept*
- together β and v are called the *parameters*
- interpretation: β_i is amount $\hat{y}$ changes when x_i increases by one with all x_j fixed

House price regression model

y: selling price (in 1000 dollars) of a house in some neighborhood, over a time period

- x_1 is the area (1000 square feet)
- x_2 is the number of bedrooms

the regression model

$$\hat{y} = 54.4 + 148.73x_1 - 18.85x_2$$

predicts the price in terms of attributes or features ($\hat{y}$ is predicted selling price)

house	x_1 (area)	x_2 (beds)	y (price)	$\hat{y}$ (prediction)
1	0.846	1	115.00	161.37
2	1.324	2	234.50	213.61
3	1.150	3	198.00	168.88
4	3.037	4	528.00	430.67
5	3.984	5	572.50	552.66

Example: house sale prices

- scatter plot shows sale prices for 774 houses in Sacramento
- in practice, regression models for house prices use many regressors and are more accurate

Regression model in matrix form

given N features (examples, samples) $x^{(1)}, \ldots, x^{(N)}$ and outcomes $y^{(1)}, \ldots, y^{(N)}$

- associated predictions are $\hat{y}^{(i)} = (x^{(i)})^T\beta + v$
- write as

$$\hat{y}^{\mathrm{d}} = X^T\beta + v\mathbf{1} = \left[\begin{array}{c} \mathbf{1}^T \\ X \end{array}\right]^T \left[\begin{array}{c} v \\ \beta \end{array}\right]$$

  - X is feature matrix with columns $x^{(1)}, \ldots, x^{(N)}$
  - $\hat{y}^{\mathrm{d}} = (\hat{y}^{(1)}, \ldots, \hat{y}^{(N)})$ is N-vector of predictions

- vector of *prediction errors* or *residuals*

$$r^{\mathrm{d}} = y^{\mathrm{d}} - \hat{y}^{\mathrm{d}} = y^{\mathrm{d}} - X^T\beta - v1$$

$y^{\mathrm{d}} = (y^{(1)}, \ldots, y^{(N)})$ is N-vector of responses (true outcomes if known)

Outline

- linear and affine functions
- Taylor approximation
- regression model
- **linear equations**
- linear dynamical systems

Systems of linear equations

set (system) of m linear equations in n variables $x_1, \ldots, x_n$:

$$\begin{aligned} A_{11}x_1 + A_{12}x_2 + \cdots + A_{1n}x_n &= b_1 \\ A_{21}x_1 + A_{22}x_2 + \cdots + A_{2n}x_n &= b_2 \\ &\vdots \\ A_{m1}x_1 + A_{m2}x_2 + \cdots + A_{mn}x_n &= b_m \end{aligned}$$

- can express compactly as $Ax = b$
- A_{ij} are the *coefficients*; A is the *coefficient matrix*
- b is called the *right-hand side*
- may have no solution, a unique solution, infinitely many solutions

Classification

- under-determined if $m < n$ (A is wide; less equations than unknowns)
- square if $m = n$ (A is square)
- over-determined if $m > n$ (A is tall; more equations than unknowns)

Example: Polynomial interpolation

- polynomial of degree at most $n-1$ with coefficients $x_1, x_2, \ldots, x_n$:

$$p(t) = x_1 + x_2 t + x_3 t^2 + \cdots + x_n t^{n-1}$$

- fit polynomial to m given points $(t_1, y_1), \ldots (t_m, y_m)$

- this is a system of linear equations:

$$Ax = \begin{bmatrix} 1 & t_1 & \cdots & t_1^{n-1} \\ 1 & t_2 & \cdots & t_2^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & t_m & \cdots & t_m^{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

  here A is the *Vandermonde matrix*

Example: Recovery of function from derivative

consider finding a function $v(t)$ from its second derivative $-g(t)$ on interval $[0, 1]$

- this problem arises in many applications such as the heat equation in one variable
- for any v with $-\frac{d^2v}{dt^2}(t) = g(t)$, the function $w(t) = v(t) + \alpha + \beta t$ has the same second derivative for any constants α and β
- to fix these constants we need two additional constraints
- we assume $v(0) = v(1) = 0$
- this yields a differential equation, $-\frac{d^2v}{dt^2}(t) = g(t)$, with boundary conditions

- we subdivide the interval $[0,1]$ into N subintervals of size $h = 1/N$ each

- define $v_k = v(kh)$ and $g_k = g(kh)$ for $k = 0, 1, \ldots, N$

- discrete approximation of $-\frac{d^2v}{dt^2}(t) = \lim_{h\to 0} \frac{v(t+h)-2v(t)+v(t-h)}{h^2} = g(t)$ is

$$-\frac{d^2v}{dt^2}(kh) \approx -\frac{v_{k+1} - 2v_k + v_{k-1}}{h^2} = g_k, \quad k = 1, 2, \ldots, N-1$$

- for boundary conditions $v(0) = 0$, $v(1) = 0$, we write $v_0 = 0$, $v_N = 0$

- rewriting the equations in matrix-vector form, we get $Av = g$, where

$$v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{N-1} \end{bmatrix}, \quad g = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_{N-1} \end{bmatrix}, \quad A = \frac{1}{h^2}\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}$$

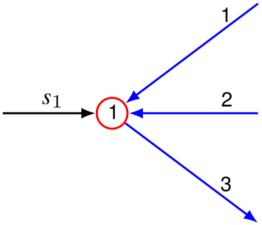
Example: Diffusion system

diffusion system is a model that arises in physics to describe *flows* and *potentials*

Flows

- consider a directed graph with n nodes and m edges
- f_j is flow across edge j (*e.g.*, electricity, heat, energy, or mass)
- s_i is source flow at node i
- in diffusion system, flows satisfy *flow conservation* (sum of flows equal zero)
- example:

$$f_1 + f_2 - f_3 + s_1 = 0$$

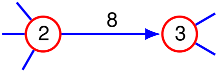


- flow conservation at every node is $Af + s = 0$ where A is the incidence matrix

Potentials

- with node i we associate a potential v_i (*e.g.*, temperature in thermal model, voltage in an electrical circuit)
- flow on an edge is proportional to the potential difference across its adjacent nodes $r_j f_j = v_k - v_l$ where r_j is *resistance* of edge j
- example:

$$r_8 f_8 = v_2 - v_3$$



- edge flow equations: $Rf = -A^T v$, where $R = \mathrm{diag}(r)$ is called *resistance matrix*

Diffusion model

$$\begin{bmatrix} A & I & 0 \\ R & 0 & A^T \end{bmatrix} \begin{bmatrix} f \\ s \\ v \end{bmatrix} = 0$$

- a set of $n + m$ homogeneous equations in $m + 2n$ variables
- to these underdetermined equations we can specify some entries of f, s, v

Outline

- linear and affine functions
- Taylor approximation
- regression model
- linear equations
- **linear dynamical systems**

Linear dynamical system

sequence of n-vectors $x_1, x_2, \ldots$

$$x_{t+1} = A_t x_t, \quad t = 1, 2, \ldots$$

- A_t are $n \times n$ *dynamics matrices*
- t denotes time or period
- x_t is *state* at time t; sequence is called (state) *trajectory*
- x_t is *current state*, x_{t-1} is *previous state*, x_{t+1} is *next state*
- examples: x_t represents
  - mechanical variables (positions or velocities)
  - age distribution in a population
  - portfolio that changes daily
- system is *time-invariant* if $A_t = A$ (doesn't depend on time)
- for time-invariant system $x_{t+\ell} = A^\ell x_t$ (A^ℓ propagates the state forward ℓ times)

Linear dynamical system

(Linear) K-Markov model

$$x_{t+1} = A_1 x_t + A_2 x_{t-1} + \cdots + A_K x_{t-K+1}, \quad t = K, K+1, \ldots$$

- next state depends on current state and $K-1$ previous states
- also known as *auto-regressive model*
- for $K = 1$, this is the standard linear dynamical system $x_{t+1} = A x_t$

Linear dynamical system with input

$$x_{t+1} = A_t x_t + B_t u_t + c_t, \quad t = 1, 2, \ldots$$

- u_t is an *input* m-vector (or exogenous variable)
- B_t is $n \times m$ *input matrix*
- c_t is *offset* (or noise)
- for fixed A, B, and $c_t = 0$,

$$x_{t+\ell} = A^{\ell} x_t + A^{\ell-1} B u_t + A^{\ell-2} B u_{t+1} + \cdots + B u_{t+\ell-1}$$

Linear dynamical system with state feedback

$$x_{t+1} = Ax_t + Bu_t, \quad t = 1, 2, \ldots\ldots$$

- the input u_t is something we can manipulate, *e.g.*, the control
- in state feedback control, input u_t is a linear function of the state,

$$u_t = Kx_t$$

  where K is the $m \times n$ state-feedback gain matrix
- with state feedback, we have

$$x_{t+1} = Ax_t + Bu_t = (A + BK)x_t, \quad t = 1, 2, \ldots$$

- recursion is the *closed-loop system* ($x_{t+1} = Ax_t$ is open-loop system)
- matrix $A + BK$ is called the *closed-loop dynamics matrix*
- widely used in many applications (we will see methods for choosing K)

Example: Population distribution

model the evolution of age distribution in some population over time by linear dynamical system

- $x_t \in \mathbb{R}^{100}$ gives population distribution in year $t = 1, \ldots, T$
- $(x_t)_i$ is the number of people with age $i - 1$ in year t (say, on January 1)
  - total population in year t is $\mathbf{1}^T x_t$
  - number of people age 70 or older in year t is $(0_{70}, \mathbf{1}_{30})^T x_t$
- birth rate $b \in \mathbb{R}^{100}$
  - b_i is average number of births per person with age $i - 1$
- death (or mortality) rate $d \in \mathbb{R}^{100}$
  - d_i is the portion of those aged $i - 1$ who will die this year (we'll take $d_{100} = 1$)
- b and d can vary with time, but we'll assume they are constant

let's find next year's population distribution x_{t+1} (ignoring immigration)

Population distribution dynamics

- number of 0-year-olds next year is total births this year:

$$(x_{t+1})_1 = b^T x_t$$

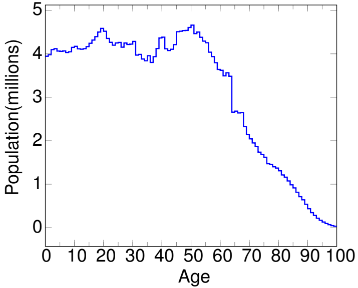
- no. of i-year-olds next year is no. of $(i-1)$-year-olds this year, minus deaths:

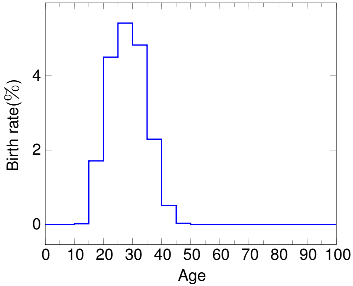
$$(x_{t+1})_{i+1} = (1 - d_i)\,(x_t)_i\,, \quad i = 1, \ldots, 99$$

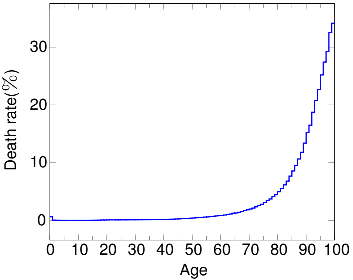
- hence, $x_{t+1} = Ax_t$, where

$$A = \begin{bmatrix} b_1 & b_2 & \cdots & b_{99} & b_{100} \\ 1-d_1 & 0 & \cdots & 0 & 0 \\ 0 & 1-d_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1-d_{99} & 0 \end{bmatrix}$$

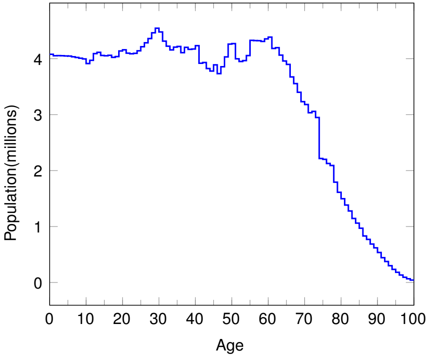
- we can use this model to predict the total population in future population distribution, birth, and death rates in the U.S. in 2010







predicting U.S. 2020 distribution from 2010 (ignoring immigration) with initial value x_1 given by the 2010 age distribution



Example: Epidemic dynamics

4-vector x_t gives proportion of population in 4 infection states

- *susceptible:* $(x_t)_1$ can acquire the disease the next day
- *infected:* $(x_t)_2$ have the disease
- *recovered:* $(x_t)_3$ had the disease, recovered, now immune
- *deceased:* $(x_t)_4$ had the disease, and unfortunately died

Example: $x_t = (0.75, 0.10, 0.10, 0.05)$ means in day t

- 75% of the population is susceptible
- 10% is infected
- 10% is recovered and immune
- 5% has died from the disease

Model assumption: suppose over each day

- 5% of susceptible acquires the disease (95% remain susceptible)
- 1% of infected dies
- 10% of infected recovers with immunity
- 4% of infected recover without immunity (*i.e.*, become susceptible)
- 85% remain infected
- 100% of immune and dead people remain in their state

Epidemic dynamics as linear dynamical system

- susceptible portion in the next day

$$(x_{t+1})_1 = 0.95\,(x_t)_1 + 0.04\,(x_t)_2$$

  - $0.95\,(x_t)_1$ is susceptible individuals from today, who did not become infected,
  - $0.04\,(x_t)_2$ is infected individuals today who recovered without immunity

- infected portion in the next day

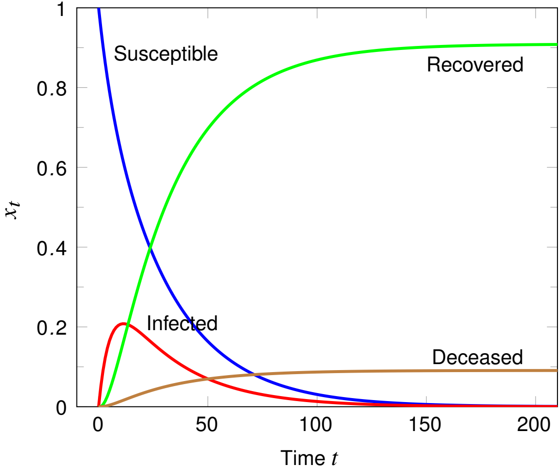
$$(x_{t+1})_2 = 0.85\,(x_t)_2 + 0.05\,(x_t)_1$$

  - first term counts those who are infected and remain infected
  - second term counts those who are susceptible and acquire disease

- using similar arguments for $(x_{t+1})_3$ and $(x_{t+1})_4$, we get

$$x_{t+1} = \begin{bmatrix} 0.95 & 0.04 & 0 & 0 \\ 0.05 & 0.85 & 0 & 0 \\ 0 & 0.10 & 1 & 0 \\ 0 & 0.01 & 0 & 1 \end{bmatrix} x_t$$

simulation from $x_1 = (1, 0, 0, 0)$

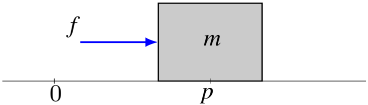


Example: Motion of a mass

- linear dynamical systems can be used to (approximately) describe the motion of many mechanical systems
- for example, an airplane (that is not undergoing extreme maneuvers)
- we describe the simplest case: a single mass moving in 1-D

Motion of mass dynamics

$$m\frac{d^2p}{d\tau^2}(\tau) = -\eta\frac{dp}{d\tau}(\tau) + f(\tau)$$



- $m > 0$ is the mass
- $f(\tau)$ is the external force acting on the mass at time τ
- $\eta > 0$ is the drag coefficient
- introducing the velocity of the mass, $v(\tau) = dp(\tau)/d\tau$, we can write

$$\frac{dp}{d\tau}(\tau) = v(\tau), \quad m\frac{dv}{d\tau}(\tau) = -\eta v(\tau) + f(\tau)$$

Discretization

- let $h > 0$ be a small time interval (called the *sampling interval*)
- define the continuous quantities 'sampled' at multiples of h seconds

$$p_k = p(kh), \quad v_k = v(kh), \quad f_k = f(kh)$$

- we now use the approximations

$$\frac{dp}{d\tau}(kh) \approx \frac{p_{k+1} - p_k}{h}, \quad \frac{dv}{d\tau}(kh) \approx \frac{v_{k+1} - v_k}{h}$$

- this leads to the (approximate) equations

$$\frac{p_{k+1} - p_k}{h} = v_k, \quad m\frac{v_{k+1} - v_k}{h} = f_k - \eta v_k$$

- using state $x_k = (p_k, v_k)$, we write this as

$$x_{k+1} = \begin{bmatrix} 1 & h \\ 0 & 1 - h\eta/m \end{bmatrix} x_k + \begin{bmatrix} 0 \\ h/m \end{bmatrix} f_k, \quad k = 1, 2, \ldots$$

References and further readings

- S. Boyd and L. Vandenberghe. *Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares,* Cambridge University Press, 2018.
- L. Vandenberghe. *EE133A lecture notes,* University of California, Los Angeles. (`http://www.seas.ucla.edu/~vandenbe/ee133a.html`)