1 Pandemic Modeling - SIRD model

- S(t): # susceptible of contracting the infection at time t;
- I(t): # infected at time t;
- R(t): # recovered from the disease up to time t;
- D(t): the cumulative # that dued due to disease up to time t.
- $\bar{I}(t)$: # of reported infected at tim t; ($\bar{S}(t)$, $\bar{R}(t)$, and $D\bar{(}t)$ are similar)
- P is the total population (assumed to be a constant)
- ω: the percentage of the population that is capable of infection to the disease.
- q is the fraction of the population that is susceptible (assumed to be a constant)

The interactions can be represented schematically as

$$\begin{array}{rcl} S+I & \longrightarrow^{\beta} & I+I \\ I & \longrightarrow^{\gamma} & R \\ I & \longrightarrow^{\nu} & D \end{array}$$

1.1 Continuous Time Model

Time t is a continuous variable. The differential equation model is given by

$$\begin{array}{rcl} \dot{S} & = & -\beta\frac{SI}{S+I} \\ \dot{I} & = & \beta\frac{SI}{S+I} - \gamma I - \nu I \\ \dot{R} & = & \gamma I \\ \dot{D} & = & \nu I \end{array} \tag{1}$$

$$0 \le \omega \le 1$$
$$\alpha \ge 1$$
$$S = \omega Pop - I - R - D$$
$$S = \alpha\bar{S}$$
$$I = \alpha\bar{I}$$
$$R = \alpha\bar{R}$$
$$D = \alpha\bar{D}$$
$$\alpha\bar{S} = \omega Pop - \alpha\bar{I} + \alpha\bar{R} + \alpha\bar{D}$$
$$\bar{S} = \frac{\omega}{\alpha}Pop - \bar{I} + \bar{R} + \bar{D}$$
$$q = \frac{\omega}{\alpha}$$
$$\bar{S} = qPop - \bar{I} + \bar{R} + \bar{D}$$

The ordinary differential equations part we can rewrite.

$$\begin{bmatrix} \dot{S} \\ \dot{I} \\ \dot{R} \\ \dot{D} \end{bmatrix} = \begin{bmatrix} -\frac{SI}{S+I} & 0 & 0 \\ \frac{SI}{S+I} & -I & -I \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \beta \\ \gamma \\ \nu \end{bmatrix}$$

It should be clear that the matrix can be solved using actual (S, I, R, D) or reported $(\bar{S}, \bar{I}, \bar{R}, \bar{D})$ numbers. For the remainder of this paper S, I, R, D will be used as a proxy for $\bar{S}, \bar{I}, \bar{R}, \bar{D}$.

Let's denote $\boldsymbol{x}(t) = \begin{bmatrix} S(t) \\ I(t) \\ R(t) \\ D(t) \end{bmatrix}$ (the state), $\boldsymbol{\theta} = \begin{bmatrix} \beta \\ \gamma \\ \nu \end{bmatrix}$ (the parameters of the model) and $\Phi = \begin{bmatrix} -\frac{SI}{S+I} & 0 & 0 \\ \frac{SI}{S+I} & -I & -I \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}$ (the dynamics). Then (1) is rewriten as

$$\dot{\boldsymbol{x}}(t) = \Phi(t; q)\,\boldsymbol{\theta} \tag{2}$$

subject to $qPop = S + I + R + D$.

1.2 Discrete Time Model

Time t is a discrete variable (i.e., 1,2,3,4...).f The difference version of (2) is given by

$$\Delta \boldsymbol{x}(t) = \Phi(t; q)\,\boldsymbol{\theta} \tag{3}$$

where $\Delta \boldsymbol{x}(t) = \boldsymbol{x}(t+1) - \boldsymbol{x}(t)$ and subject to $qP = S + I + R + D$. Note that the parameters $\boldsymbol{\omega}$ are different from the parameters of the continuous model.

$$\begin{bmatrix} S(t+1) - S(t) \\ I(t+1) - I(t) \\ R(t+1) - R(t) \\ D(t+1) - D(t) \end{bmatrix} = \begin{bmatrix} -\frac{S(t)I(t)}{S(t)+I(t)} & 0 & 0 \\ \frac{S(t)I(t)}{S(t)+I(t)} & -I(t) & -I(t) \\ 0 & I(t) & 0 \\ 0 & 0 & I(t) \end{bmatrix} \begin{bmatrix} \beta \\ \gamma \\ \nu \end{bmatrix}$$

If β, γ, and ν are functions of time instead of constants, the discrete time model can be written as

$$\begin{bmatrix} S(t+1) - S(t) \\ I(t+1) - I(t) \\ R(t+1) - R(t) \\ D(t+1) - D(t) \end{bmatrix} = \begin{bmatrix} -\frac{S(t)I(t)}{S(t)+I(t)} & 0 & 0 \\ \frac{S(t)I(t)}{S(t)+I(t)} & -I(t) & -I(t) \\ 0 & I(t) & 0 \\ 0 & 0 & I(t) \end{bmatrix} \begin{bmatrix} \beta(t) \\ \gamma(t) \\ \nu(t) \end{bmatrix}$$

Then for each t, $\beta(t)$, $\gamma(t)$, and $\nu(t)$ can be solved by the Moore-Penrose pseudoinverse $\theta(t) = \Phi^{\dagger}(t; q)\Delta \boldsymbol{x}(t)$.

1.3 Parameter Estimation of the Discrete Time Model using Basis Functions

For a given q and each $t = 0, 1, 2, \ldots, T$

- data available - for $t = 0, 1, 2, \ldots, T$ we have $I(t), R(t), D(t)$.
- $S(t) = qP - I(t) - R(t) - D(t)$
- compute $\Delta \boldsymbol{x}(t)$ and $\Phi(t; q)$

For each $t > 0$, and a forgetting factor $w \in (0, 1]$ (used $w = 0.9$ and $w = 0.7$)

- build

$$\boldsymbol{y} = \begin{bmatrix} \sqrt{w^{T-1}}\Delta \boldsymbol{x}(1) \\ \sqrt{w^{T-2}}\Delta \boldsymbol{x}(2) \\ \vdots \\ \sqrt{w^{T-t}}\Delta \boldsymbol{x}(t) \\ \vdots \\ \Delta \boldsymbol{x}(T) \end{bmatrix}$$

- build

$$A = \begin{bmatrix} \sqrt{w^{T-1}}\Phi(1; q) \\ \sqrt{w^{T-2}}\Phi(2; q) \\ \vdots \\ \sqrt{w^{T-t}}\Phi(t; q) \\ \vdots \\ \Phi(T; q) \end{bmatrix}$$

- minimize the cost function

$$\min_{\theta} f(q, \boldsymbol{\theta}) = \min_{\theta} \frac{1}{T}\|\boldsymbol{y} - A\boldsymbol{\theta}\|_2^2 + \lambda\|\theta\|_1 \tag{4}$$

  with $\lambda = 10$.

Note that

$$\frac{1}{T}\|\boldsymbol{y} - A\boldsymbol{\theta}\|_2 = \frac{1}{T}\sum_{t=1}^{T} w^{T-t}\|\Delta \boldsymbol{x}(t) - \Phi(t; q)\,\boldsymbol{\theta}\|_2^2$$

captures how much each measurement $\Delta \boldsymbol{x}(t)$ differs from the prediction $\Phi(t; q)\,\boldsymbol{\theta}$(if we were to use the parameters $\boldsymbol{\theta}$) by having the most recent measurements worth more; $\lambda\|\theta\|_1$ is a regularitation term on the parameters.

Equation (4) can be solved in python using

```
from sklearn import linear_model
clf = linear_model.Lasso(alpha=lambda/2, fit_intercept=false, positive=true)
clf.fit(A, y)
print(clf.coef_)
```

1.4 Model with Time Varying Parameters

The parameters β, γ, ν are time varying. Let represent them as a linear combination of basis functions. Let $\{b_i(t)\}$, $\{g_i(t)\}$, $\{m_i(t)\}$ be three sets of basis functions and define

$$
\begin{aligned}
B(t) &= [b_1(t), \ldots, b_{n_1}(t)] \\
\boldsymbol{\beta} &= [\beta_1, \ldots, \beta_{n_1}] \\
G(t) &= [g_1(t), \ldots, g_{n_2}(t)] \\
\boldsymbol{\gamma} &= [\gamma_1, \ldots, \gamma_{n_2}] \\
M(t) &= [m_1(t), \ldots, m_{n_3}(t)] \\
\boldsymbol{\nu} &= [\nu_1, \ldots, \nu_{n_1}]
\end{aligned}
$$

then (3) can be expressed as

$$
\begin{aligned}
\dot{\boldsymbol{x}}(t) = \begin{bmatrix} \dot{S} \\ \dot{I} \\ \dot{R} \\ \dot{D} \end{bmatrix} &= \begin{bmatrix} -b_1\frac{SI}{S+I} & \cdots & -b_{n_1}\frac{SI}{S+I} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ b_1\frac{SI}{S+I} & \cdots & b_{n_1}\frac{SI}{S+I} & -g_1 I & \cdots & -g_{n_2} I & -m_1 I & \cdots & -m_{n_2} I \\ 0 & \cdots & 0 & g_1 I & \cdots & g_{n_2} I & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 & m_1 I & \cdots & m_{n_2} I \end{bmatrix} \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_{n_1} \\ \gamma_1 \\ \vdots \\ \gamma_{n_2} \\ \nu_1 \\ \vdots \\ \nu_{n_3} \end{bmatrix} \\
&= \begin{bmatrix} -B\frac{SI}{S+I} & \mathbf{0} & \mathbf{0} \\ B\frac{SI}{S+I} & -GI & -MI \\ \mathbf{0} & GI & 0 \\ \mathbf{0} & \mathbf{0} & MI \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\gamma} \\ \boldsymbol{\nu} \end{bmatrix} \\
\dot{\boldsymbol{x}}(t) &= \boldsymbol{\Phi}(t)\boldsymbol{\theta}
\end{aligned}
$$

1.4.1 Parameter Estimation

For a given q and each $t = 0, 1, 2, \ldots, T$

- data available - for $t = 0, 1, 2, \ldots, T$ we have $I(t), R(t), D(t)$.
- $S(t) = qP - I(t) - R(t) - D(t)$
- Pick the basis functions b_i, g_i, m_i and the number of the basis functions used n_1, n_2, n_3 in approximating $\beta(t), \gamma(t)$ and $\nu(t)$ respectively
- compute $\Delta\boldsymbol{x}(t)$ and $\boldsymbol{\Phi}(t; q)$

For each $t > 0$, and a forgetting factor $w \in (0, 1]$ (used $w = 0.9$ and $w = 0.7$)

- build

$$
\boldsymbol{y} = \begin{bmatrix} \sqrt{w^{T-1}}\Delta\boldsymbol{x}(1) \\ \sqrt{w^{T-2}}\Delta\boldsymbol{x}(2) \\ \vdots \\ \sqrt{w^{T-t}}\Delta\boldsymbol{x}(t) \\ \vdots \\ \Delta\boldsymbol{x}(T) \end{bmatrix} \tag{5}
$$

- build

$$A = \begin{bmatrix} \sqrt{w^{T-1}}\mathbf{\Phi}(1;q) \\ \sqrt{w^{T-2}}\mathbf{\Phi}(2;q) \\ \vdots \\ \sqrt{w^{T-t}}\mathbf{\Phi}(t;q) \\ \vdots \\ \mathbf{\Phi}(T;q) \end{bmatrix}$$

- minimize the cost function

$$\min_{\theta} f(q,\boldsymbol{\theta}) = \min_{\theta} \frac{1}{T}\|\boldsymbol{y} - A\boldsymbol{\theta}\|_2^2 + \lambda\|\theta\|_1 \tag{6}$$

  with $\lambda = 10$.

Note that

$$\frac{1}{T}\|\boldsymbol{y} - A\boldsymbol{\theta}\|_2 = \frac{1}{T}\sum_{t=1}^{T} w^{T-t}\|\Delta\boldsymbol{x}(t) - \Phi(t;q)\boldsymbol{\theta}\|_2^2$$

captures how much each measurement $\Delta\boldsymbol{x}(t)$ differs from the prediction $\Phi(t;q)\boldsymbol{\theta}$(if we were to use the parameters $\boldsymbol{\theta}$) by having the most recent measurements worth more; $\lambda\|\theta\|_1$ is a regularitation term on the parameters.

Equation (6) can be solved in python using

```
from sklearn import linear_model
clf = linear_model.Lasso(alpha=lambda/2, fit_intercept=false , positive=true)
clf.fit(A, y)
print(clf.coef_)
```

1.5 Model with Time Varying Parameters

Using basis functions $\beta(t)$, $\gamma(t)$, and $\nu(t)$ should be determined as well as $\Phi(t;q)$, I, R, and D known from the reported data, and S determined from the chosen value of q. Then future values can be approximated

$$0 \le t \le T_p$$
$$0 \le T \le T_p$$

Then for all t in range $T+1, T+2, \ldots, T_p$

$$\boldsymbol{x}(t+1) = \Delta\boldsymbol{x}(t) + \boldsymbol{x}(t)$$
$$\boldsymbol{x}(t+1) = \Phi(q;t)\theta(T) + \boldsymbol{x}(t)$$

To normalize the prediction a similar method can be used. Let $p(t,t_0)$ be the predicted model state at time t using parameters from time t_0.

$$p(t,t_0) = \Phi(q;t-1)\theta(t_0) + \boldsymbol{x}(t-1)$$

For all t in range $T+1, T+2, \ldots, T_p$

$$\boldsymbol{x}(t) = \frac{1}{2^T}p(t,0) + \sum_{\tau=1}^{T}\frac{p(t,\tau)}{2^{T-\tau+1}}$$

In practice this can be solved by

For all t in range $T+1, T+2, \ldots, T_p$

$x(t) = p(t, 0)$

For all τ in range $1, 2, 3, \ldots, T$

$x(t) \leftarrow \frac{1}{2}x(t) + \frac{1}{2}p(t, \tau)$