Early Detection of COVID-19 Hotspot Counties Using Data Science

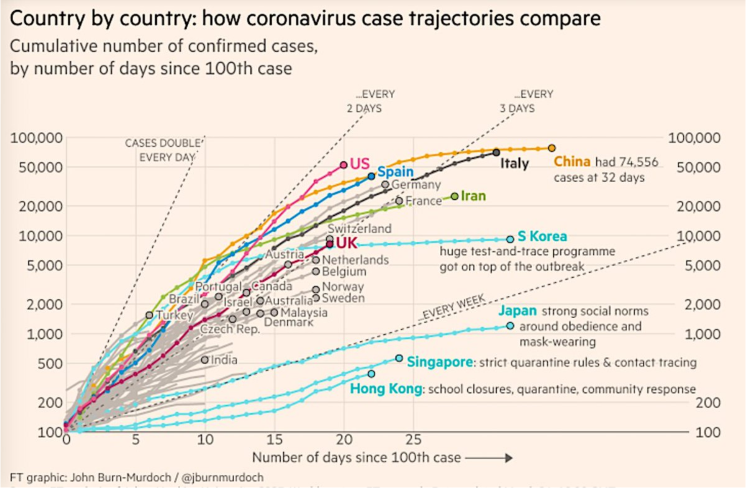
Zhaowei She [1] Zilong Wang [1] Jagpreet Chhatwal [2] Turgay Ayer [1]

[1]Georgia Institute of Technology

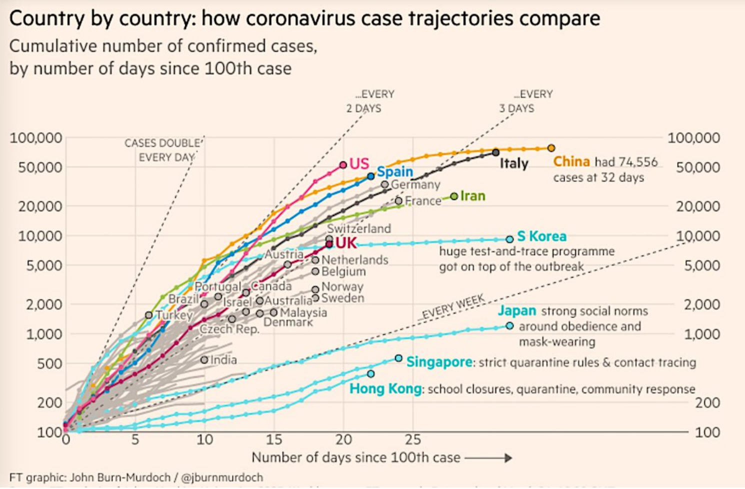
[2]Harvard Medical School and MGH

June 19th, 2020

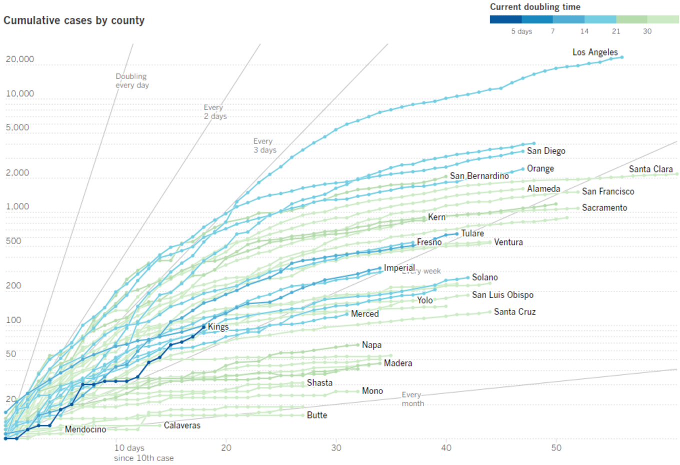
Flatten the Curve!



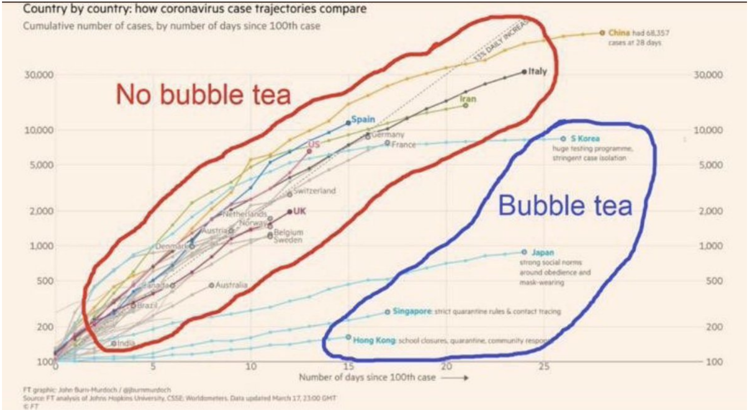
Separate the Flattened Ones from the Others!



How Do We Know Which Curves are Flattened?



We Need a Classification Algorithm!



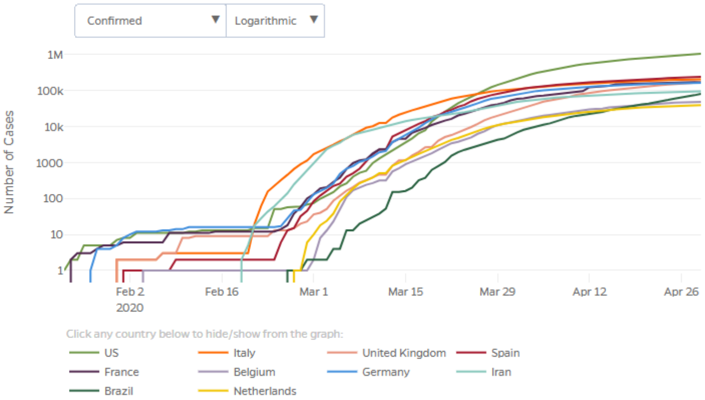
A Statistical Model of the “Curves"

A Two-Parameter Exponential Growth Model (Heroy, 2020):

$$I_{t,c} = e^{r_c(t-t_{0,c})} + \epsilon_{t,c}$$

- *Dependent Variable* $(I_{t,c})$: The cumulative number of infected individuals in location c at time $t - t_{0,c}$ days
- *Independent Variable* (t): The current time
- *Parameters* $(r_c, t_{0,c})$:
  - r_c: Exponential growth rate
  - $t_{0,c}$: The outbreak time of location c

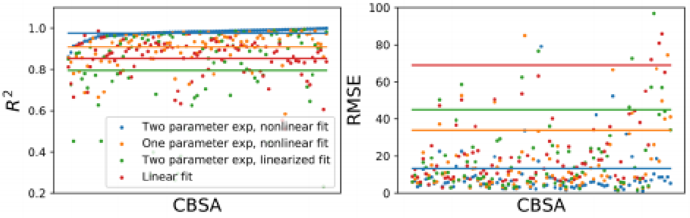
A Statistical Model of the “Curves"



$$I_{t,c} = e^{r_c(t - t_{0,c})} + \epsilon_{t,c}$$

A Statistical Model of the “Curves"

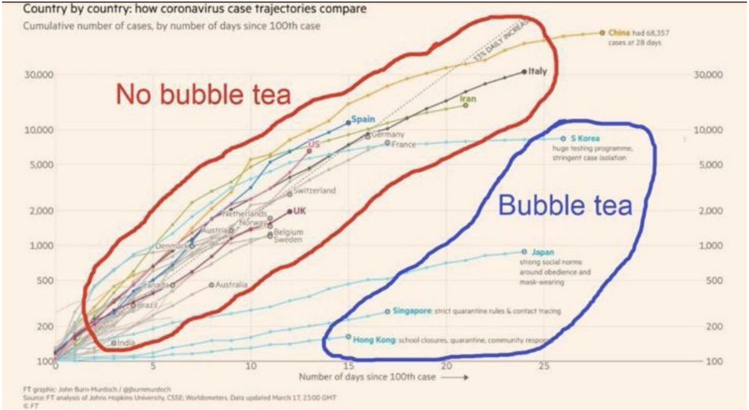
Shown to have good fit R^2 and $RMSE$



Goodness of Fit in terms of R^2 and Root Mean Square Error (RMSE)

$$I_{t,c} = e^{r_c(t-t_{0,c})} + \epsilon_{t,c}$$

Classification Based On the “Curves"



“Curves": r_c and $t_{0,c}$ in

$$I_{t,c} = e^{r_c(t-t_{0,c})} + \epsilon_{t,c}$$

Classification Based On the “Curves"

$$\underset{\substack{(R_i,T_{0,i})\in\mathbb{R}^2\forall i\in I\\ x_i\in\{0,1\}^{|C|}\forall i\in I}}{Min}\{m_c(r_c,t_{0,c})\}^T\{m_c(r_c,t_{0,c})\}$$

$$s.t.\ \sum_{i\in I} x_i = 1_{|C|\times 1}$$

$$x_i(c)=1 \Rightarrow \begin{bmatrix} r_c \\ t_{0,c}\end{bmatrix} = \begin{bmatrix} R_i \\ T_{0,i}\end{bmatrix} \quad \forall i\in I\ \forall c\in C,$$

where

$$m_c(r_c,t_{0,c}) := \partial_{\begin{bmatrix} r_c \\ t_{0,c}\end{bmatrix}} \mathbb{E}[(I_{t,c}-e^{r_c(t-t_{0,c})})^2].$$

Statistical Justifications

Counterargument: Why not fit a curve to every county, then sort and cut off based off their rates?

- *Autocorrelation*: Time series violate exogeneity assumption for regression
- *Sparse Data*: County level data too sparse, potential overfitting
- *No clear cutoffs*: Debatable heuristics we want statistical guarantees

Preprocessing

Feature Engineering: Convert forecasting into regression problem. For each county c's time series data:

- *Lag Variables*: Add rolling 7 day mean to smoothen data and capture autocorrelations

$$\bar{I}_{t,c} := \frac{1}{7}\sum_{i=1}^{7} I_{t-i,c}$$

- *Detrending*: Apply diff log operation to consecutive lag variables

$$\hat{I}_{t,c} := \ln(\bar{I}_{t,c}) - \ln(\bar{I}_{t-1,c})$$

- *Time delayed embedding*: Auto-regress on past variables + richer feature space (forecast next 7 days with past 14 days)

$$X_{t,c} = \begin{bmatrix} \hat{I}_{t,c} & \hat{I}_{t-1,c} & \dots & \hat{I}_{t-13,c} \\ \hat{I}_{t+1,c} & \hat{I}_{t,c} & \dots & \hat{I}_{t-12,c} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{I}_{t+6,c} & \hat{I}_{t+5,c} & \dots & \hat{I}_{t-7,c} \end{bmatrix}$$

Stage 1: Recursive Partitioning

Greedy Approximation to MICP Formulation

- *Recursive Partitioning*: Optimal Cluster Assignment is NP-hard (intractable), we split based on the sorted features of the time embedded matrix ala CART / C4.5 Decision Trees
- *Split Criterion*: The criterion we used is the conditional sum of Weighted Mean Absolute Percentage Errors (wMAPE) of the validation set using the XGBoost subroutine on each child
- *Cross Validation and Backtesting*: To determine the optimal depth (number of clusters) we cross validated against historical data in a sliding window approach

Stage 2: Forecasting Each Cluster

Recovering the Results

- *Model Selection:* The model with the lowest wMAPE from cross validation was selected

$$\text{wMAPE}(\text{Ground Truth} = \vec{y}, \text{Predicted} = \hat{y}) := \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{\sum_{i=1}^{n} |y_i|}$$

- *Cluster Prediction:* Ever county in the same cluster is then fitted with the same exponent growth model

Cross Validation Errors

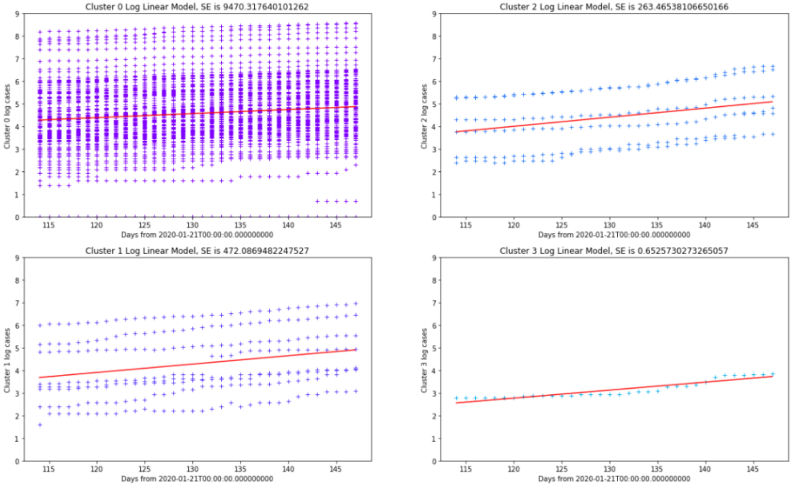
Note: wMAPA$:= 1-$wMAPE

Cross Validation Results		
Fold	wMAPA (%)	Best Depth
1	99.69	2
2	99.51	2
3	99.65	2
4	99.70	2
5	99.69	2
6	99.74	2
7	99.67	3
8	**99.83**	**2**
9	99.77	2
10	99.80	2
11	99.49	2
12	99.67	2

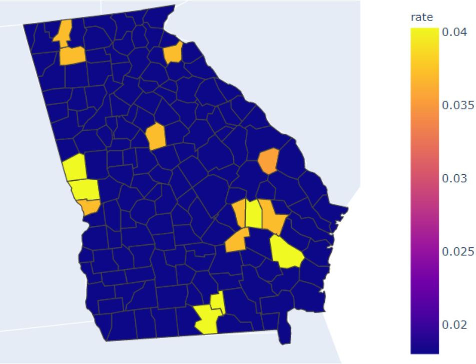
Runtime ≈ 10 minutes

Data Source: NYTimes-COVID-19 Data

Cluster Plots of County Cases



Choropleth of Forecasted Cases



Highest Risk:
Jenkins County

Higher Risk:
Harris County, Lanier County,
Lowndes County, Tombs County,
Troup County, Wayne County

High Risk:
Franklin County, Gordon County,
Jasper County, Jeff Davis County,
Montgomery County, Muscogee County,
Tattnall County, Whitfiled County

Low Risk:
the rest

Algorithm Properties and Output

- *Shape Matching:* From the scatter plots, our method clusters time series together by general shape and trend, emulating Discrete Time Warping (DTW) methods
- *Hierarchical Clustering*: Unlike classical Agglomerative Clustering using DTW, which constructs clusters bottom up, ours construct it top down, making our model more interpretable
- *Efficient Computationally*: 13 folds of cross validated (with each split running 2 x 1000 boosted trees) models of large transformed data took less than 10 minutes on a notebook with a single Intel i7-9750H CPU @ 2.60GHz and 16 GB of RAM

Comments on Application

Impact

- *Highly Interpretable:* Counties are clearly partitioned into stratified risk tiers
- *Reasonable Advanced Warning*: We are able to forecast which counties are potential hotspots 7 days in advance
- *Surprising Results*: Some high risk counties such as Montgomery have historically very low cases, which would have been missed out with simple prediction forecasting

Reference

Heroy, S. (2020). Metropolitan-scale covid-19 outbreaks: how similar are they? *arXiv preprint arXiv:2004.01248*.