CSE 8803 EPI: Data Science for Epidemiology, Fall 2022

Lecturer: B. Aditya Prakash
Scribe: Thomas Lang, Sagar Badlani, Jeffrey Chang

October 25, 2022
Lecture 15 : Forecasting (I)
Lecture 16 : Forecasting (II)

1 Lecture Summary

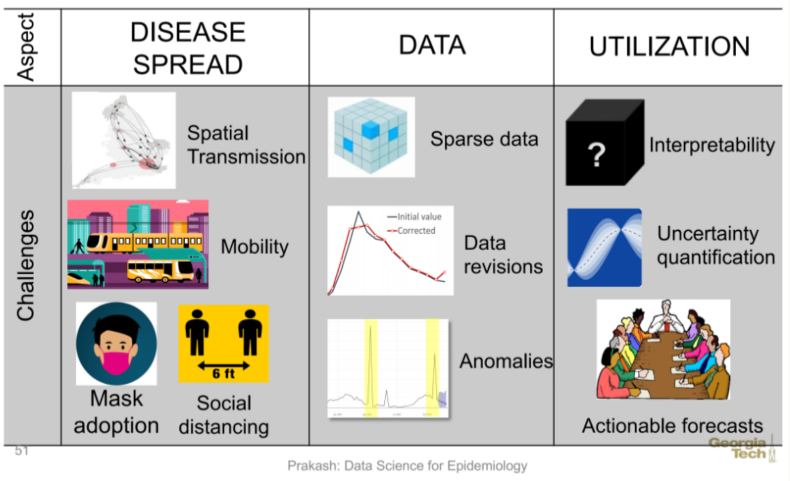
This lecture covers the use of Statistical, ML and AI models for Epidemiological Forecasting. From relatively simple models (regression) to complex models (recurrent neural networks), the lecture provides a foundation for understanding how machine learning is applied to Epidemiological problems as well as provides examples of state of the art applications of ML to specific problems.

2 Statistical, ML and AI Models for Forecasting

Statistical, ML and AI models use historical data to learn patterns that can be used to predict different outcomes. In the application of this class, these models use prior epidemic data for forecast peak times, peak intensities, reproduction numbers and more. In this approach, one finds the best model, out of a selection of possible approaches, that performs the best using the given historical data. Generally, these functions are trained by inputting historical data into a model and comparing the generated output to the ground truth to adjust model parameters and increase accuracy. A high-level representation of this is shown below:

$$min \sum_{i=1}^{T} \mathcal{L}(f(x_i) - y_i)$$

Where $f(x_i)$ is the prediction of the model given an input x_i and y_i is the ground truth. Then we optimize parameters that minimize the loss function $\mathcal{L}$. At the end of training, a subset of the dataset that isn't used for training is used to determine how effective the model is.



Aspect	DISEASE SPREAD	DATA	UTILIZATION
Challenges	Spatial Transmission Mobility Mask adoption Social distancing	Sparse data Data revisions Anomalies	Interpretability Uncertainty quantification Actionable forecasts

Figure 1: ML Model Considerations

It is worth noting that ML models don't try to model the mechanics of epidemics (unlike mechanistic models). It is not recommended to try to use ML models to model the long term transmission of diseases, as there is limited data and mechanistic models perform better. Future lectures on hybrid models will detail how to combine these approaches. That being said, ML approaches have the advantage of supporting a variety of training data (including time series, text, and image data). One challenge with these models is that epidemic data can be sparse or require data revisions (which can decrease model accuracy). Additionally, these models are used to inform public health policy decisions so they must be interpretable. For example, a model that forecasts cases might be used to implement public health policies (like social distancing). So models that can be explained can make justify public health actions. The figure above expands on factors that must be considered when using ML models. In the following sections, we will explore different models.

2.1 Regression Models

Regression models assume a linear relationship between input features and results. These inputs can be high-dimensional and multi-modal (including data from prior epidemic cases, search trends, word occurrences in text, features from satellite images, and more). Due to the temporal nature of epidemics (e.g. daily case counts) autoregressive models are popular. In this approach, past values of epidemics are used to predict future values:

$$y_t = \sum_{j=1}^{p} \phi_j y_{t-j} + \phi_0 + e$$

Where y_t is the value being predicted, y_{t-j} are the past values, e is error and ϕ_i are the parameters to learn. In the next section, we will explore specific examples of auto-regressive models.

2.1.1 ARGO

ARGO is one example of an AutoRegression model that uses Google Search Trend data of common symptoms (e.g. coughing, fever, etc.) to predict flu cases. The approach of using a linear regression model on flu search symptom trends was proposed in the Google Flu Trends (GFT). ARGO differs from GFT by learning parameters for each symptom and also using ILI data from the current season up to the current time iteration [7]. An overview of this approach is summarized below:

$$y_t = \mu_y + \sum_{j=1}^{N} \alpha_j y_{t-j} + \sum_{i=1}^{K} \beta_i X_{i,t} + e$$

Where the first summation learns parameters for the ILI data from the current season up until the current time-step and the second summation learns parameters for each search trend.

2.1.2 ARGO2

ARGO2 is an extension of ARGO that predicts regional and national influenza like illness (ILI) rates. This applies the original ARGO methodology to predict ILI values for each region using search trends and historical ILI data. Then, it uses a multi-variate gaussian to pool regional predictions to estimate national ILI rates. This approach models the differences of epidemic spread in different regions to provide a more nuanced and accurate national estimate. Because of this more incremental approach, this national ILI estimation is more accurate.

2.1.3 Flu Forecasting based on Surrogate Data

The paper Flu Forecasting based on Surrogate Data emphasizes how data from various sources can be aggregated using an autoregressive linear model [3]. In this paper, the author proposes using various data sources including weather data, search trends, restaurant reservation information, twitter text data and current season ILI data for future ILI predictions. This multi-modal approach shows how the combination of many data sources can result in more accurate methods (which is only possible using machine learning models).

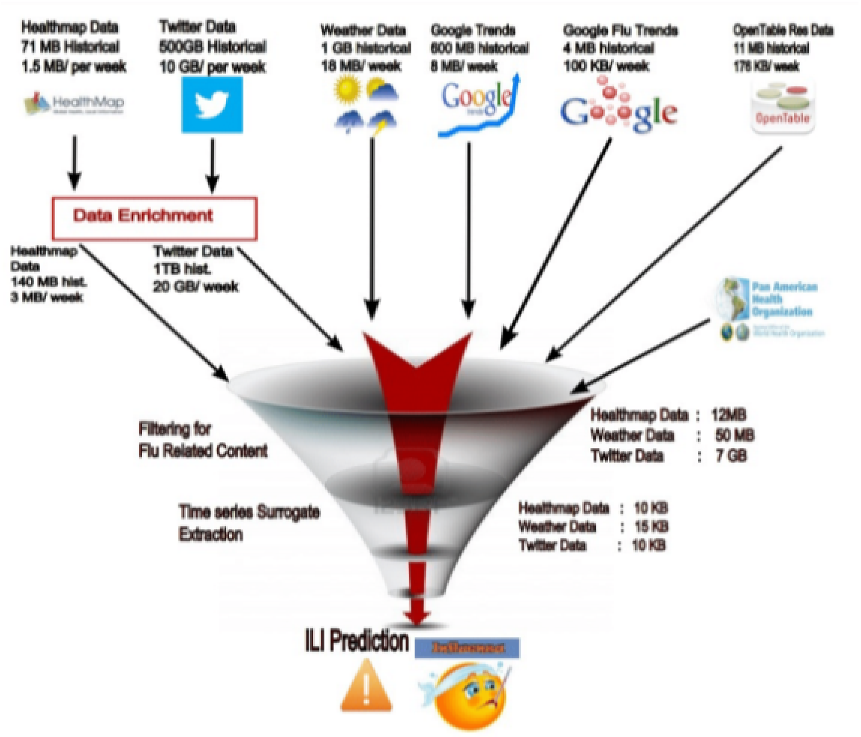


Figure 2: Flu Forecasting

2.1.4 Ensemble of ML Models

The core idea of using an Ensemble of ML models is to train multiple models using historical data and combine their outputs to get a better result. The outputs of each model can be combined using different approaches including Matrix Factorization Based Regression, Nearest Neighbor Based Regression and Matrix Factorization Regression using Nearest Neighbor embedding. The result of the combination of models is a meta-model that overcomes the limitation of a single model but using the results of multiple models.

2.2 Language Models

There is a large source of online text data that can be used to drive epidemiology models. One simple example of this is search query data. Many models analyze how searches for common disease symptoms can indicate an increase in disease cases. This is a relatively simple example of text based data. More complex examples might include extracting relevant information from social media posts (like Twitter tweets). Regardless of how the text data is collected and analyzed, the results can be used as inputs into auto-regressive models to improve predictions. This extra dimension can improve model accuracy when data is limited (e.g. when disease testing isn't possible or expensive).

2.2.1 Using Tweets to Forecast the H1N1 Pandemic.

This research used words in tweets to infer the epidemiological states of Twitter users via a HFSTM temporal model. Using this information, the researchers were able to create a mechanistic model that accurately models how people get sick and recover from a sickness sometimes better than Google Flu Trends approaches. This indicates that some of the limitations of search trends can be overcame using a different method that is also text-based [4].

2.2.2 Cross Lingual Word Embeddings For Transfer Learning.

In the field of natural language processing, a embedding for a word or phrase is a multi-dimensional vector. Similar words will have embeddings that are relatively similar to eachother (small distance in the embedded vector space) whereas different words will have different embedings (large distance in embedded the vector space). In this paper, the author uses word embeddings to find search queries that are related to disease symptoms. Using this temporal search trend data, the researchers trained a model for predicting ILI rates for a country with ILI ground-truth data available. Then, the researchers showed that using embeddings for search trends in different languages could be used by the same model to predict ILI rates. This approach was used to demonstrate how models trained in one region with ILI data can be applied to other countries with search trends in different languages and no ILI data [8].

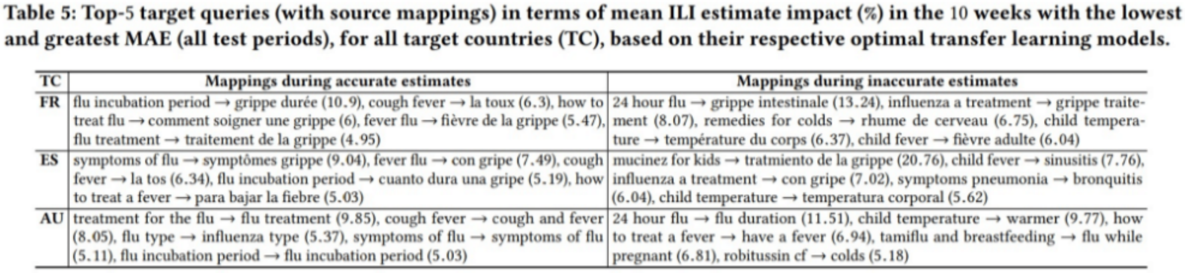


Table 5: Top-5 target queries (with source mappings) in terms of mean ILI estimate impact (%) in the 10 weeks with the lowest and greatest MAE (all test periods), for all target countries (TC), based on their respective optimal transfer learning models.

TC	Mappings during accurate estimates	Mappings during inaccurate estimates
FR	flu incubation period → grippe durée (10.9), cough fever → la toux (6.3), how to treat flu → comment soigner une grippe (6), fever flu → fièvre de la grippe (5.47), flu treatment → traitement de la grippe (4.95)	24 hour flu → grippe intestinale (13.24), influenza a treatment → grippe traitement (8.07), remedies for colds → rhume de cerveau (6.75), child temperature → température du corps (6.37), child fever → fièvre adulte (6.04)
ES	symptoms of flu → symptômes grippe (9.04), fever flu → con gripe (7.49), cough fever → la tos (6.34), flu incubation period → cuanto dura una gripe (5.19), how to treat a fever → para bajar la fiebre (5.03)	mucinez for kids → tratmiento de la grippe (20.76), child fever → sinusitis (7.76), influenza a treatment → con gripe (7.02), symptoms pneumonia → bronquitis (6.04), child temperature → temperatura corporal (5.62)
AU	treatment for the flu → flu treatment (9.85), cough fever → cough and fever (8.05), flu type → influenza type (5.37), symptoms of flu → symptoms of flu (5.11), flu incubation period → flu incubation period (5.03)	24 hour flu → flu duration (11.51), child temperature → warmer (9.77), how to treat a fever → have a fever (6.94), tamiflu and breastfeeding → flu while pregnant (6.81), robitussin cf → colds (5.18)

Figure 3: Search Trend Language Transfer

2.3 Vision Models

There is some recent work that use satellite images of locations that are sensitive to disease outbreak (e.g hospital parking lots) to measure the severity of an outbreak of a disease.

2.3.1 Satellite images to detect Flu outbreaks

In this paper, researchers used satellite images of parking lots near hospitals to estimate the number of vehicles. Using this information, and the hospital occupancy rates as truth data, the researchers trained a linear regression model to estimate weekly ILI rates [2]. Other approaches combine parking lot data with search trend and ILI counts as additional features as well.

2.4 Neural Models

Neural Models (Deep Learning) capture non-linear patterns in high dimensional data with minor assumptions. They are generally flexible and generalize to complex domains, but require large amounts of data. A current deep-learning approach to epidemiology problems is to use sequential models. These models use sequential data (e.g. time series, or words in a sentence) where the output of the last step is used as the input to the current step. This approach captures long-range patterns. Some popular examples of such models are recurrent neural networks and transformers. A high-level overview of how recurrent neural networks work is shown in the figure below.

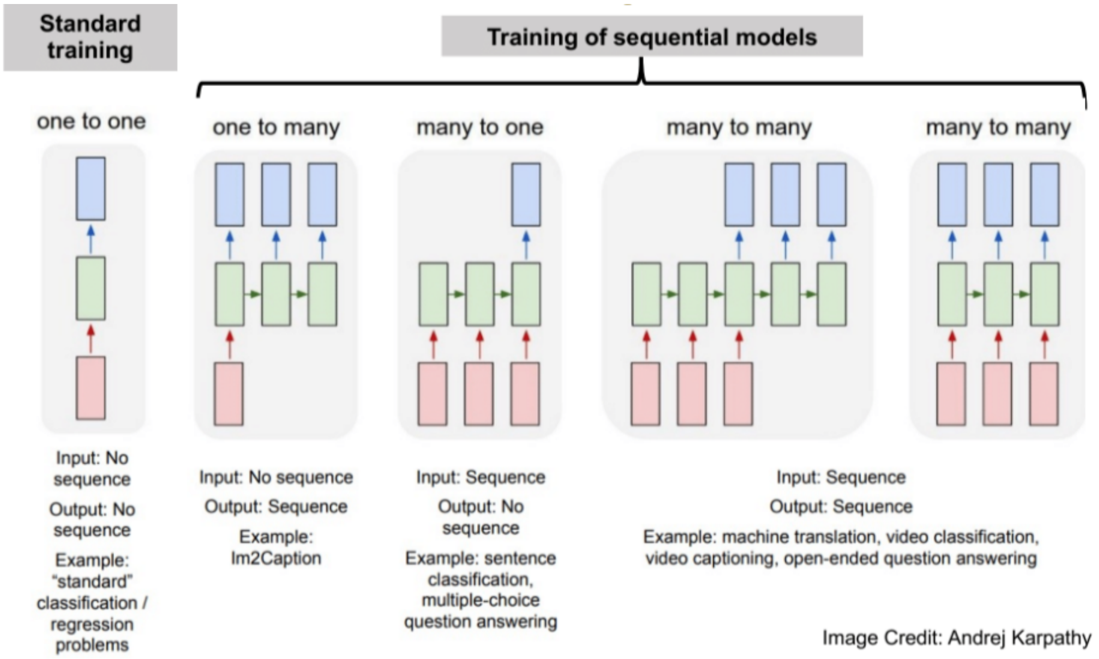
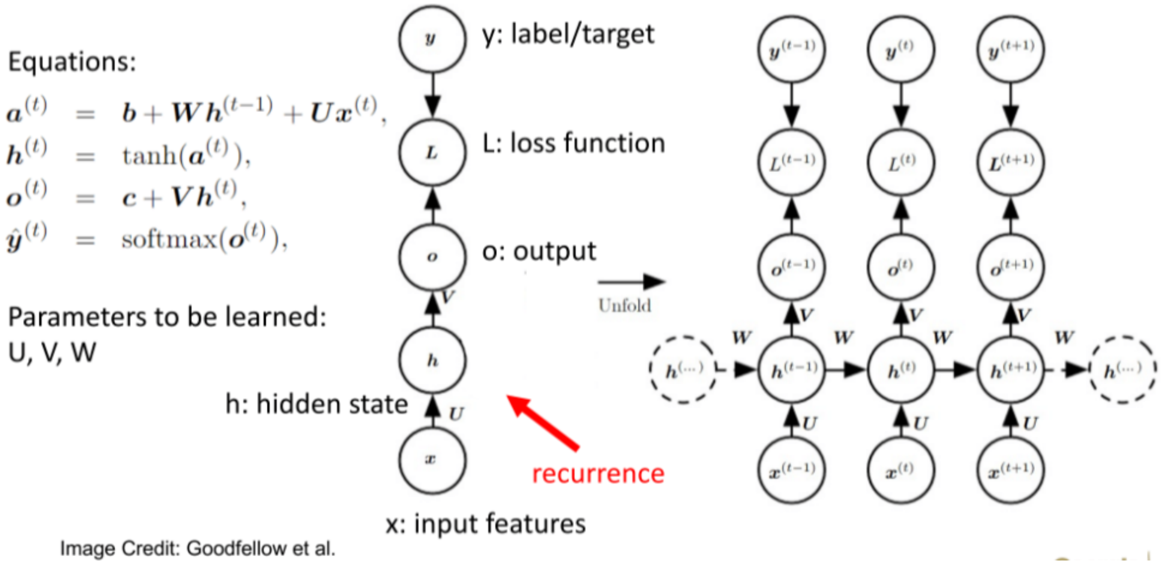

Standard training
Training of sequential models
one to one
one to many
many to one
many to many
many to many
Input: No sequence
Output: No sequence
Example: "standard" classification / regression problems
Input: No sequence
Output: Sequence
Example: Im2Caption
Input: Sequence
Output: No sequence
Example: sentence classification, multiple-choice question answering
Input: Sequence
Output: Sequence
Example: machine translation, video classification, video captioning, open-ended question answering
Image Credit: Andrej Karpathy

Figure 4: Sequential Models

In a sequential model, each logical block contains multiple logical functions. One function learns the parameters used for the input of the block and another function learns the parameters for the input of the prior block's output. A different function learns the parameters for generating the block's output and a final function learns the parameters that will be used to create the input for the next sequential block. A more detailed overview of this is shown in the figure below:



Equations:

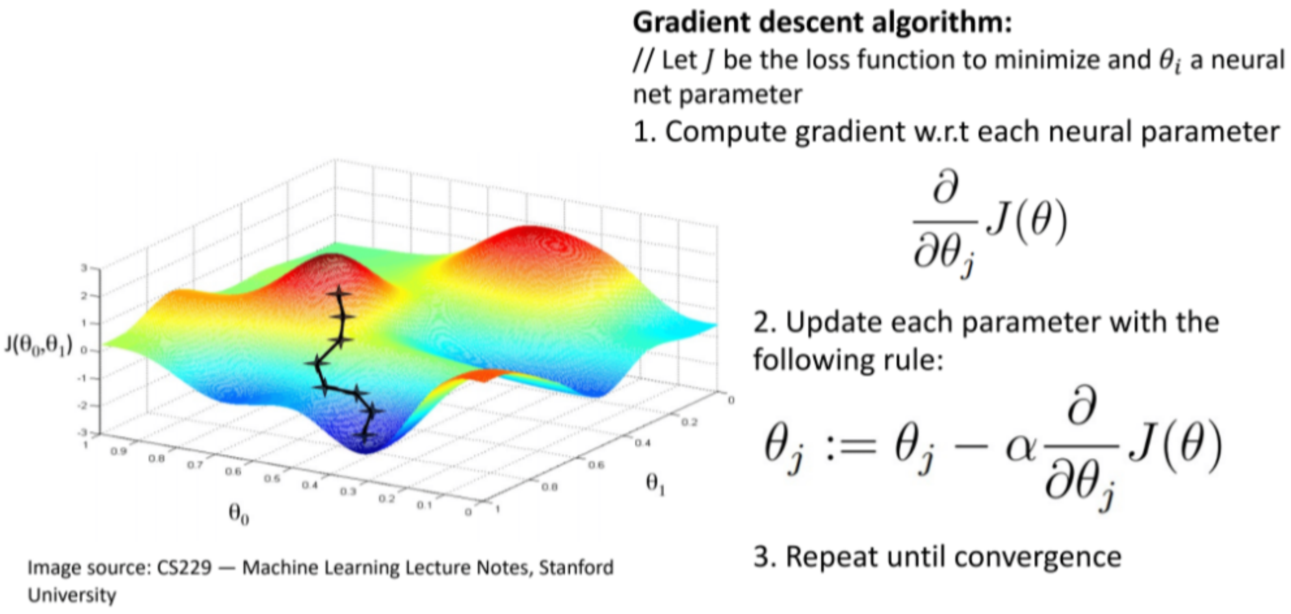
$$\begin{aligned} \boldsymbol{a}^{(t)} &= \boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{U}\boldsymbol{x}^{(t)}, \\ \boldsymbol{h}^{(t)} &= \tanh(\boldsymbol{a}^{(t)}), \\ \boldsymbol{o}^{(t)} &= \boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}^{(t)}, \\ \hat{\boldsymbol{y}}^{(t)} &= \operatorname{softmax}(\boldsymbol{o}^{(t)}), \end{aligned}$$

Parameters to be learned:
U, V, W

Image Credit: Goodfellow et al.

Figure 5: Detailed overview of RNNs

As with all machine learning models, the model's output is compared to the ground-truth data of the given example to determine how the correctness of the model. This is done using a loss function. Some common examples of loss functions are mean squared error and cross-entropy loss. To optimize the model's parameters, the derivative of the loss of a given example is calculated. This derivative is known as the gradient which is then used to update each parameter (depending on how much each parameter contributed to the loss error). This is repeated until additional iterations don't increase the model's accuracy. This optimization is known as gradient descent. An example of this is shown in the figure below:



Gradient descent algorithm:

// Let J be the loss function to minimize and θ_i a neural net parameter

1. Compute gradient w.r.t each neural parameter

$$\frac{\partial}{\partial \theta_j} J(\theta)$$

2. Update each parameter with the following rule:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

3. Repeat until convergence

Image source: CS229 — Machine Learning Lecture Notes, Stanford University

Figure 6: Overview of Gradient Descent

One major issue with gradient descent in recurrent neural networks is the idea of ”vanishing gradients”. The main idea behind this problem is that the gradients of early parts of long sequential inputs will become basically zero when the loss of the model is calculated. This essentially means that the model doesn’t learn how to remember early inputs which limits the patterns that RNN models can learn. This limitation motivated LSTMs, a variation of RNNs, that use ”gates” which decides which data in a sequence is important to keep or okay to throw away. This allows the model to keep relevant early information when necessary.

One final augmentation of Recurrent Neural Networks is the use of the Attention Mechanism. This allows models to pay attention to only parts of an input sequence that are most relevant to predict the output of an input. Essentially, attention captures the relevance of an RNN block’s state in determining the output. Inputs that don’t contribute to determining the output won’t have high attention which signals to the model to focus on the other RNN blocks instead. The attention weights $a_{i,j}$ are calculated using the logic below:

$$a_{i,j} = softmax(score(h^d_{i-1}, h^e_j)\forall j \epsilon e)$$

Where the score function captures the relevance of each encoder hidden state to the decoder state:

$$score(h^d_{i-1}, h^e_j) = h^d_{i-1} * h^e_j$$

A visual example of attention is shown in the figure below:

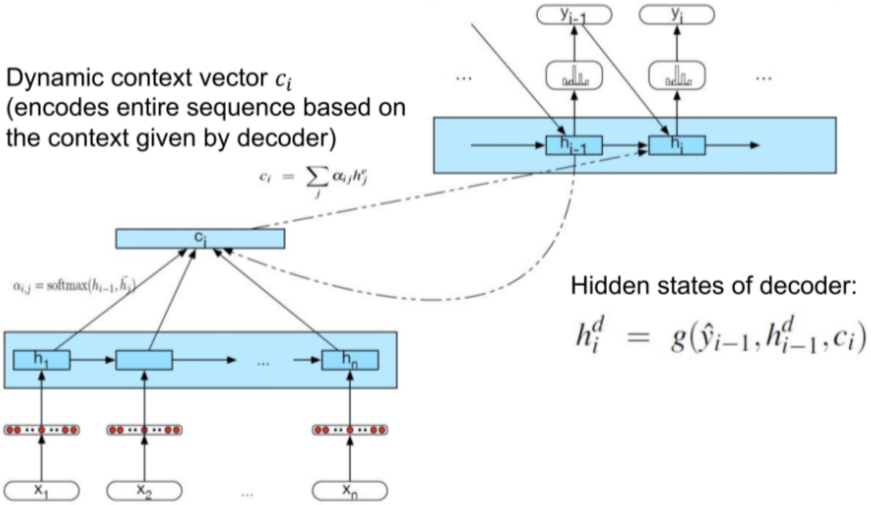

Dynamic context vector c_i
(encodes entire sequence based on the context given by decoder)
$c_i = \sum_j \alpha_{ij} h^e_j$
$\alpha_{i,j} = \mathrm{softmax}(h_{i-1}, \bar{h}_j)$
Hidden states of decoder:
$h^d_i = g(\hat{y}_{i-1}, h^d_{i-1}, c_i)$
y_{i-1}
y_i
h_{i-1}
h_i
c_i
h_1
h_n
x_1
x_2
x_n

Figure 7: Attention Overview

Transformers are recurrent neural networks that use self-attention. Self-attention does not depend on the output of the model (unlike normal attention) and only depends on the input itself.

2.5 Training a model for Forecasting

When training a model for forecasting future targets, careful consideration must be taken to partition the dataset. Given a time series X = $x_1, x_2, ..., x_t$ and a time series window of length L, we can splice the dataset to get the inputs for each time step. An example is shown below:

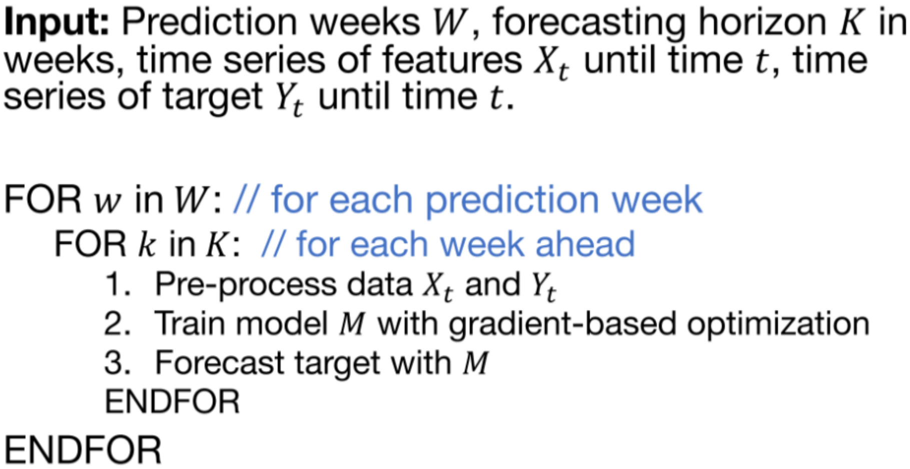
$$input = [x_1, ..., x_{L-1}, x_L] \;\; target = y_{L+k}$$

$$input = [x_2, ..., x_L, x_{L+1}] \;\; target = y_{L+1+k}$$

$$...$$

$$input = [x_{t-L-k}, ..., x_{t-1-k}, x_{t-k}] \;\; target = y_t$$

The pseudocode for training a model for real-time forecasting is shown below:



```
Input: Prediction weeks $W$, forecasting horizon $K$ in
weeks, time series of features $X_t$ until time $t$, time
series of target $Y_t$ until time $t$.

FOR $w$ in $W$: // for each prediction week
    FOR $k$ in $K$:  // for each week ahead
        1. Pre-process data $X_t$ and $Y_t$
        2. Train model $M$ with gradient-based optimization
        3. Forecast target with $M$
        ENDFOR
ENDFOR
```

Figure 8: Realtime Model Training Psuedocode

2.6 Neural Models (Continued)

The previous section details the application of general Neural Networks and Deep Learning models to epidemic forecasting. This section elaborates some deep learning approaches specific to Epidemiology. These models are designed to overcome the problem of data sparsity, improve inductive bias and interpretability, and incorporate transfer learning and the spatiotemporal nature of disease spread.

2.6.1 EpiDeep

EpiDeep [1] is one such approach that leverages the similarity with past data. This is particularly useful for seasonal diseases (eg. Flu). The model uses the historical data of the past seasons and the partially observed data for the current season to predict the future incidence, the peak time, the peak intensity and the onset. Leveraging the similarity between the current season and the past seasons is non-trivial because the current season is only observed till week t while the historical seasons are fully observed. EpiDeep overcomes this challenge by using two different clustering steps, Query Length Data Clustering and Full Length Data clustering. The first step captures the similarity between the current season and the past seasons only up to time t. The observed part of the current season is termed as the 'query', while the truncated parts of the past seasons till week t are referred to as query length historical data. Query Length Data Clustering is performed using Improved

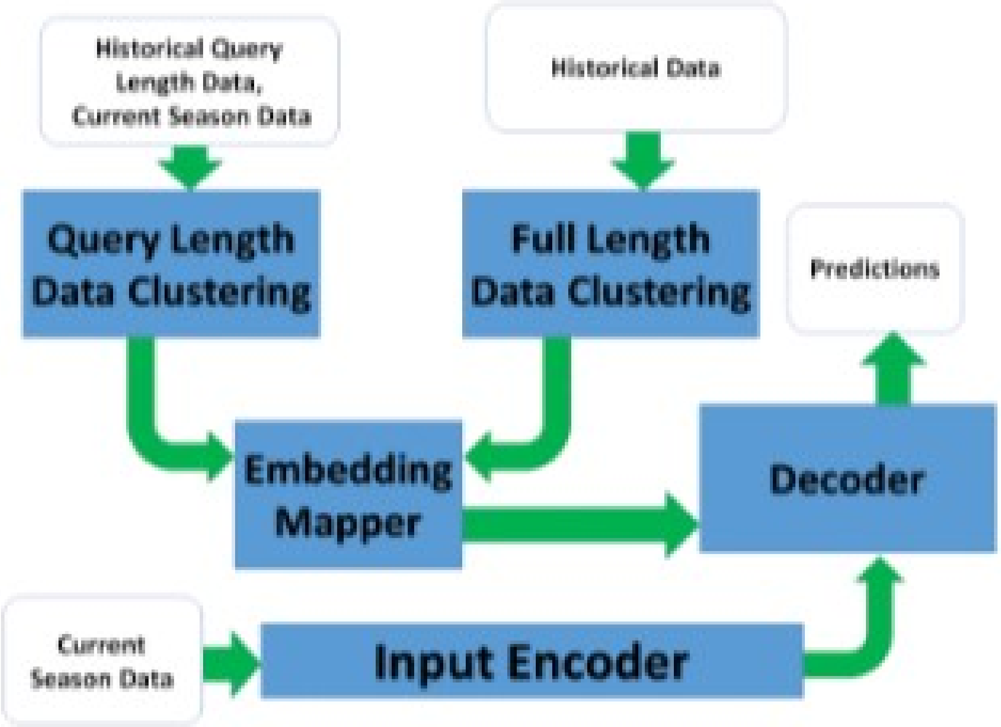
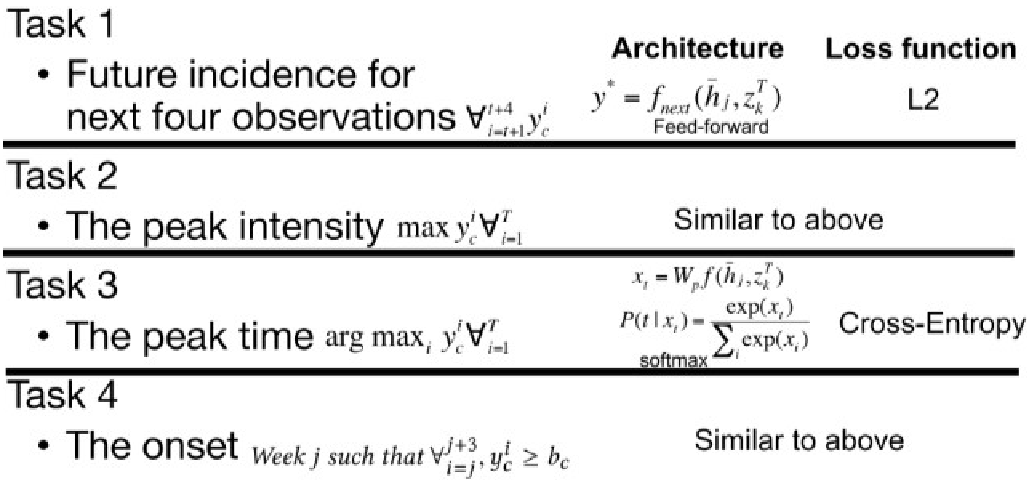

Historical Query Length Data, Current Season Data
Historical Data
Query Length Data Clustering
Full Length Data Clustering
Predictions
Embedding Mapper
Decoder
Current Season Data
Input Encoder

Figure 9: Overview of the EpiDeep Architecture

Deep Embedded clustering (IDEC) which is an auto-encoder which minimizes clustering loss. The next step involves capturing the similarities between the complete length of the historical seasons without the current season. This is done using Full Length Data clustering. A mapping function uses the embeddings from both the steps to convert the query length embedding of the current season to represent a complete season. The output of the mapping function is fed to a Decoder along with the encoded input for the current season. The role of the Decoder is to make the required predictions. The high level architecture of the EpiDeep model is given in Fig. 9.

The Input Encoder is an LSTM network that uses an attention mechanism approach to assign different weights to parts of the data. This reduces reliance on the latest data. However, this encoding suffers from data sparsity and does not generalize well. To overcome these issues, the encoded data is combined with the similarity embedding from the mapping function as explained above.

The Decoder is a feed-forward network which exploits different architectures for different prediction tasks. The Decoder uses the encoded input $\bar{h}_j$, and the embedding z_c^T to perform the predictive tasks. The summary of the different architectures is given below. The EpiDeep model outperformed the baselines by up to 40% for the US National region as well as for other HHS regions.



	Architecture	Loss function
Task 1 • Future incidence for next four observations $\forall_{i=t+1}^{t+4} y_c^i$	$y^* = f_{next}(\bar{h}_j, z_k^T)$ Feed-forward	L2
Task 2 • The peak intensity $\max y_c^i \forall_{i=1}^{T}$	Similar to above	
Task 3 • The peak time $\arg\max_i y_c^i \forall_{i=1}^{T}$	$x_t = W_p f(\bar{h}_j, z_k^T)$ $P(t \mid x_t) = \frac{\exp(x_t)}{\sum_i \exp(x_i)}$ softmax	Cross-Entropy
Task 4 • The onset $Week\ j\ such\ that\ \forall_{i=j}^{j+3}, y_c^i \geq b_c$	Similar to above	

2.6.2 Using Multiple Clustering Methods

This approach was popular during the early days of COVID-19, when data for different regions was not abundantly available. The technique involves training a single model for a set of regions. The regions are clustered based on their geographical similarity. Ensemble approaches are used to combine multiple models having different clustering strategies.

2.6.3 Inter-series attention

The past time-series of all the regions are segmented and passed through Convolutional layers to convert them to fixed embeddings. Attention-based similarity is used between the input for the current season and the embeddings of the past seasons to make predictions.

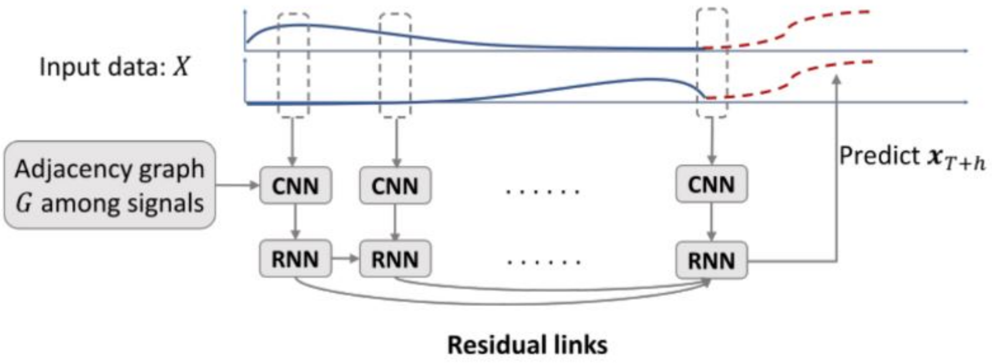
2.6.4 Transfer knowledge representations

This method is used to transfer models from a data-rich domain to a data-scarce domain. Transfer learning can also reduce the computational costs. An example includes transferring a historical model to a novel scenario. For instance, many pre-COVID flu models were transferred to analyze COVID-contaminated flu counts. However, the historical flu models were unable to adapt to the new trends and dynamics induced by COVID-19.
CALI-Net is an example where a historical flu model was fine-tuned and adapted to the new COVID-related signals like testing data, mobility information, exposure details, crowd-sourced and social media surveys. The historical flu model was combined with the COVID-ILI model which captured these signals. The historical model was steered using the Hint Loss and the Attentive imitation loss to adapt it to the new trends.

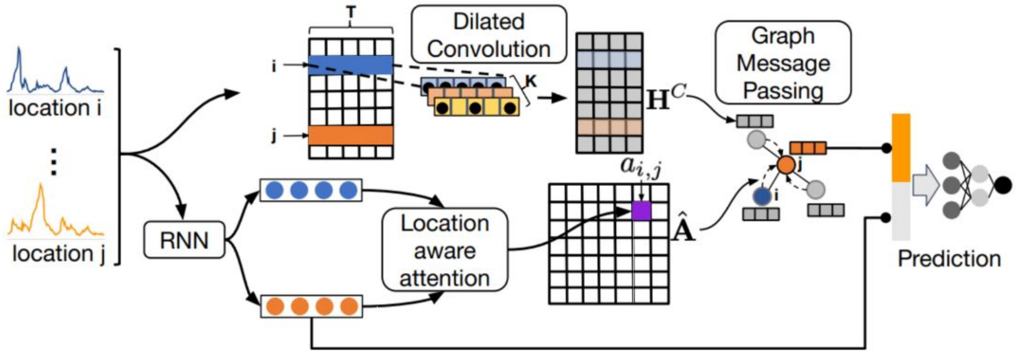
2.6.5 Incorporate Spatial Structure

When monitoring disease, pathogens will propagate to adjacent regions in a graph representation. If we use a spatial graph as a representation, and connect nearby nodes, we can track the spread of pathogens. This can be tackled in a few ways.

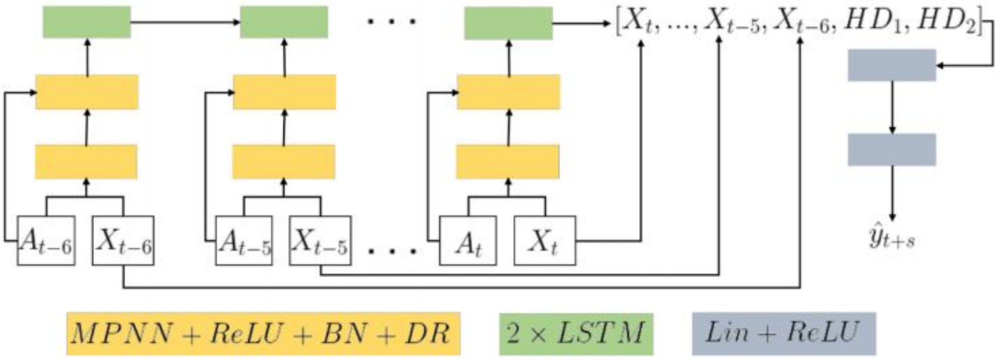
First of all, by combining CNNs and RNNs, it is possible to model the entire situation. CNNs are able to capture the regional proximity, while RNNs are able to model temporal dynamics. With residual connections, there is much better generalization to unknown scenarios.



Another possibility is using ColaGNN. This is a graph neural network that also maintains spatial structure. It uses dilated convolution for temporal modeling, so it is able to keep track of both the sptial and temporal aspects of the problem. ColaGNN is also found to be improve accuracy in making long term predictions.

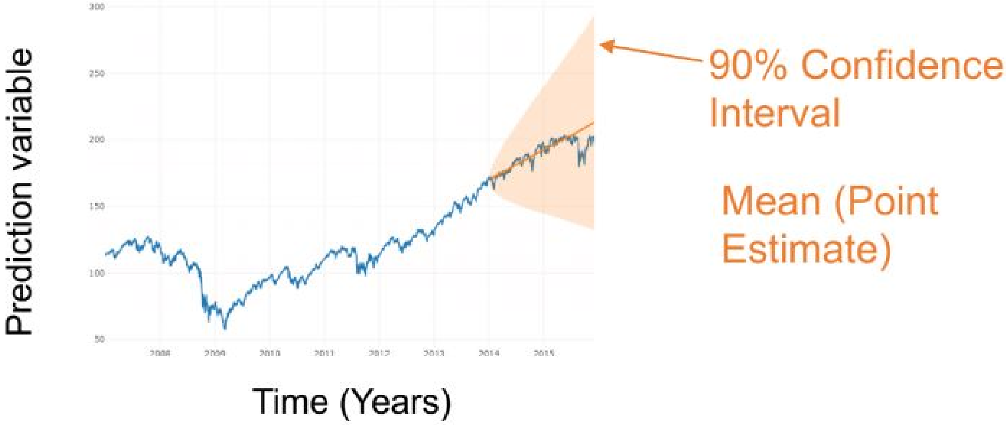


A third possibility for the issue is to use Transfer Learning with Graph Neural Networks. One can construct a graph using movement data across the regions of a country to generate a reasonably accurate graph of the situation. From there, GNN (MPNN) and LSTM can be combined to monitor the spatial and temporal aspects respectively of the question. Meta-Learning can be then used to train over different regions, which will help improve accuracy for regions with less data.



2.7 Density Estimation

With a Density Estimation Model, the goal is to directly model the forecast distribution. In other words, when given X and Y, we want to be able to predict future probabilities from this data. This generally works, as the model is able to focus on point-predictions. Additionally, the model is probabilistic, and thus can do well at conveying uncertainty and the likelihood of different outcomes.



There are a few types of density estimation functions. These include parametric, non-parametric, and neural probabilistic models. Parametric density estimation models have the parameters of distribution result from a function of the features. Non-parametric ones take a function of training datapoints and use their similarity. Finally, neural probabilistic models use deep learning to find complex, hard to notice patterns to improve their results.

2.7.1 Empirical Bayes

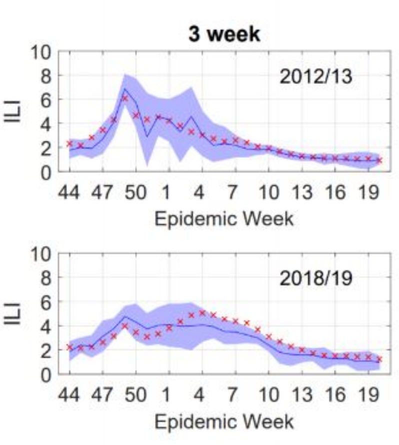
Empirical Bayes is a parametric density estimation function. It uses a probabilistic distribution to model the current season's epidemic curve. The parameters found are the peak height and week, the scaling factor of the curve, as well as the shape, which can be similar to past sequences. Using Bayesian Inference, the result for the current season can be optimized for accuracy.

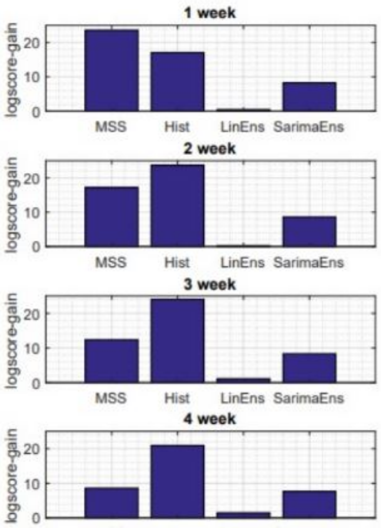
2.7.2 Delta Density

Delta Density is a non-parametric density estimation function. It uses kernel density estimation to find similarities with historical seasons. It was rather successful, and was one of the top models in the Flusight 2017 challenge.

2.7.3 Gaussian Process

Gaussian Process is a non-parametric density estimation function. It is run on the data from previous seasons, and also shows a relatively accurate confidence interval and log score over past models.



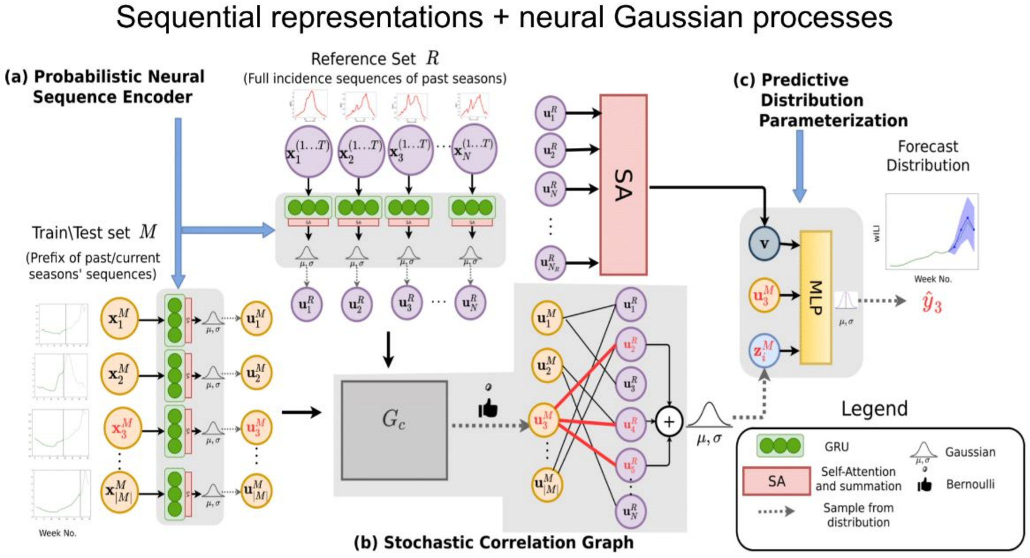


However, most statistical methods can't use relatively complex patterns that might exist, but are harder to detect with just basic statistics. Thus, when there are new conditions, they can't be expected to produce reliable forecasts on uncertainty.

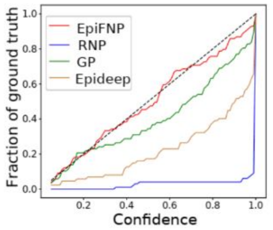
2.7.4 EpiFNP

EpiFNP is a neural non-parametric model. It improves on calibration for the forecasts from previously mentioned models. It uses a non-parametric Gaussian Process to allow for flexibility, while also looking at similarities with historical data.

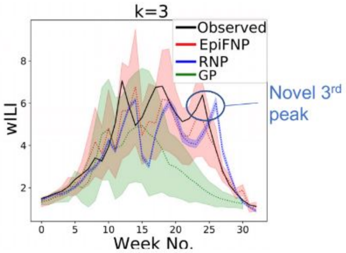
A Gaussian Process is a probability distribution over all possible functions that can fit a set of points. Parametric approaches will put knowledge into a set of parameters, while non-parametric approaches like GP will take the entire training data into account every time a prediction is made. To build predictive distributions, it is important to combine the uncertainty from all of the different areas, including the current sequence, comparisons to historical data, as well as the uncertainty of historical data.


Sequential representations + neural Gaussian processes
(a) Probabilistic Neural Sequence Encoder
Reference Set R
(Full incidence sequences of past seasons)
(c) Predictive Distribution Parameterization
Forecast Distribution
Train\Test set M
(Prefix of past/current seasons' sequences)
SA
MLP
G_c
μ, σ
$\hat{y}_3$
Week No.
wILI
(b) Stochastic Correlation Graph
Legend
GRU
Gaussian
SA
Self-Attention and summation
Bernoulli
Sample from distribution

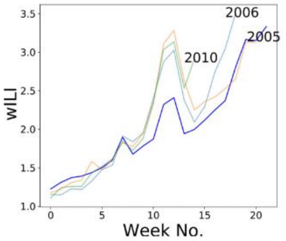
The results for EpiFNP are as follows:



Well calibrated predictions



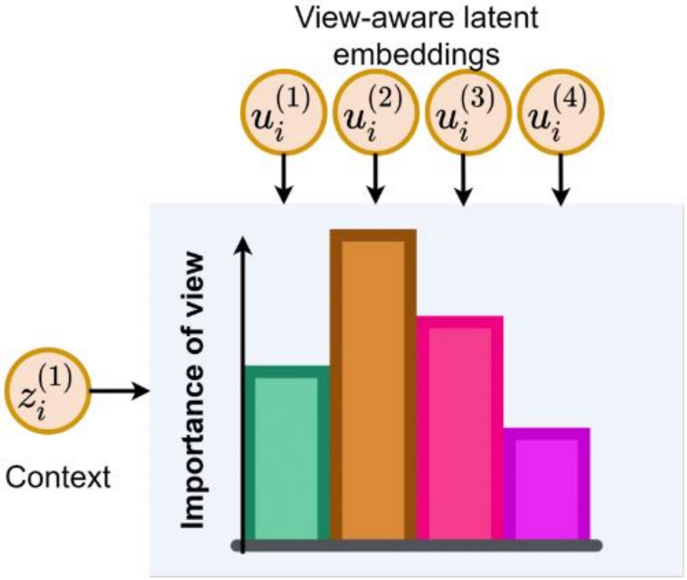
Adapt to novel patterns



Explaining predictions

It can be very challenging to use multi-view forecasting. This is because of the large number of possible inputs and variations between inputs. It is hard to capture the information and uncertainty from a number of different data sources, and to also integrate the beliefs from all of these sources. In particular, factoring in conflicting beliefs, data redundancy, variations in noise are some of the most challenging problems with multi-view forecasts.

The goal is thus to derive the importance of each view automatically based off the data. If we are able to determine this, we can combine the belief from each view with the weighted aggregation of their stochastic embeddings, which will result in a powerful overall summary. Using a latent encoder, we are able to capture information and uncertainty from each of the different views and sources. By combining them with a neural network, we can solve the issue of context specific dynamic view selection.



2.7.5 CaMuL

The CaMuL model had only an 18-30% accuracy with recards to Covid-19 and flu prediction tasks when scored with CRPS.

2.7.6 Pros/Cons List

Regarding the pros and cons of statistical models, we can summarize them as follows.

Statistical models can use a large variety of types of data directly. This includes high-dimensional data, data that might be indirectly related, and non domain-specific data regarding the dynamics of an epidemic. They are known to be excellent in multiple areas, including short-term forecasting and modern forecasting initiatives and challenges.

However, statistical models are unaware of epidemic spread mechanisms, and thus have poor long-term performance. They are also unable to evaluate potential new scenarios, and are poorly adapted to predict counterfactual scenarios. They also need to be well monitored and fine-tuned if designed for real world development, as they tend to decrease in performance over time with the data.

References

[1] B. Adhikari, X. Xu, N. Ramakrishnan, and B. A. Prakash. Epideep: Exploiting embeddings for epidemic forecasting. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 577–586, 2019.

[2] P. Butler, N. Ramakrishnan, E. O. Nsoesie, and J. S. Brownstein. Satellite imagery analysis: What can hospital parking lots tell us about a disease outbreak? *Computer*, 47:94–97, 2014.

[3] P. Chakraborty, P. Khadivi, B. L. Lewis, A. Mahendiran, J. Chen, P. Butler, E. O. Nsoesie, S. R. Mekaru, J. S. Brownstein, M. V. Marathe, and N. Ramakrishnan. Forecasting a moving target: Ensemble models for ili case count predictions. In *SDM*, 2014.

[4] L. Chen, K. S. M. Tozammel Hossain, P. Butler, N. Ramakrishnan, and B. A. Prakash. Flu gone viral: Syndromic surveillance of flu on twitter using temporal topic models. In *2014 IEEE International Conference on Data Mining*, pages 755–760, 2014.

[5] B. A. Prakash. Lecture 15: Forecasting (i). `https://www.dropbox.com/sh/jg48r4y9489oulj/AADakPzG1sH2Icax6AWnKRQPa/?preview=lecture-15.pdf`, 2022.

[6] B. A. Prakash. Lecture 16: Forecasting (ii). `https://www.dropbox.com/sh/jg48r4y9489oulj/AADakPzG1sH2Icax6AWnKRQPa/?preview=lecture-16.pdf`, 2022.

[7] S. Yang, M. Santillana, and S. Kou. Argo: a model for accurate estimation of influenza epidemics using google search data. *Proceedings of the National Academy of Sciences*, 112, 05 2015.

[8] B. Zou, V. Lampos, and I. Cox. Transfer learning for unsupervised influenza-like illness models from online search data. 02 2019.