Predicting Influenza Propagation using Network Inference and Machine Learning

Mike Chrzanowski (mc2711), Jerome Ku (jeromeku), Foon Wang Pong (ppong)
Group 6

1. Introduction

Seasonal influenza epidemics are a major public health issue, leading to millions of days of lost productivity and hundreds of thousands of deaths every year [1]. Epidemic containment is an increasingly complex issue in an ever-more connected world, necessitating means for better tracking and prediction of flu outbreaks.

The goal of our project is to predict influenza propagation in the United States. Being able to predict the spread of influenza would allow the government to take preventive measures to limit infection and anticipate the need for large-scale vaccination. To that end, we tracked the spread of influenza through hospitals in the United States between the years 2009 to 2011 as a means for predicting the spread of influenza in the general population. We trained a classifier that can accurately predict whether a hospital in the United States will observe cases of influenza in a given month by extracting features from hospital networks we constructed.

There is extensive prior work in the area of modeling epidemics [8,9]. The focus of these studies, however, is on explaining the underlying dynamics of disease spread and secondarily on prediction. Our work, in contrast, is focused primarily on prediction.

Our main contribution is the use of novel network-derived features in aiding prediction. Several networks were built from both empirical and inferred network properties. These networks included a transportation network built from ground and air transportation features, where hospitals are nodes and edges are based on Euclidean distance and flight volume between airports in proximity of hospitals. Additionally, we used network inference algorithms to derive a network from hospital inpatient diagnosis records where edges were estimated based on the the spread of disease from one hospital to another. Furthermore, to capture macro-level properties of influenza propagation, we segmented the United States into population clusters and monitored the spread of influenza within these clusters. Finally, we included non-network features such as Google Flu Trend index data as well seasonality to further improve prediction accuracy.

2. Related Work

There is extensive literature on the statistical modeling of epidemics [8,9]. The majority of the work focuses on large systems of differential equations to describe the dynamical evolution of disease spread through simulations. Most commonly, these are variants / extensions of the susceptible, infection, and removed (SIR) model.

Central to these studies is the concept of a contact network, in which nodes represent individual hosts and the edges represent potential paths of infection. From the contact network, a transmission network is realized, which represents the actual graph of disease spread.

Network structure can be estimate through various means: collection of host data, known transportation networks [7], patients in a hospital [5], or inferred from past transmission data [8]. Moreover, algorithms used for social network analysis have proven useful in studying epidemic spread [10].

The focus of these statistical models is on describing the mechanisms giving rise to the complex phenomena of epidemics. While the construction of descriptive models is a useful step in prediction, it is not always a necessary one. In fact, as has been shown through Google Flu Trends, purely data-driven methodologies with minimal

domain understanding can yield meaningfully predictive models. For example, in [11], the authors were able to pre-empt the US government's flu tracker by building simple linear models based on large-scale query data.

In our work, we combine both data-driven and domain-informed features with the overarching goal of prediction. We utilize both empirical contact networks and inferred transmission networks in crafting features. More precisely, we created a hospital contact network based on travel distance and flight volume. We also inferred a hospital transmission network based on historical propagation rates. From these networks, we derived features which along with other predictors of disease spread, were then fed into classifiers for predicting likelihood of a hospital being infected.

3. Data

3.1 NIS Dataset

We use the Healthcare Cost and Utilization Project (HCUP) National Inpatient Sample (NIS) from 2009 - 2011 as our primary data source. This data contains all discharge data from a 20% sample of all hospitals in the US, representing approximately 8 million hospital stays per year. Each medical record in the dataset treats a patient record as a unit (i.e., the same patient will have a new record for each visit) and includes ICD9 diagnosis codes, procedure codes, length of stay and additional information regarding the hospital the patient stayed at. The NIS data contains additional information including addresses, the number of beds, and also hospital type (i.e., teaching hospital). Below are summary statistics for the dataset:

Year	**2009**	**2010**	**2011**
Total hospitals	1,050	1,051	1,049
Total reported cases of influenza	30,532	5,614	15,734

Table 1. NIS dataset statistics

3.2 Supplementary Datasets

In addition to the NIS dataset, several supplementary datasets were required to create features.

Transportation / Distance: While we would have liked to use Google Maps' Distance API to estimate distances between hospitals, this was not possible due to usage limits and the large number of requests required to compute all pairwise distances between hospitals in our dataset. Instead, as we will discuss in 4.1.1.1.1, we calculated the distance between zip codes and used that as an approximation for the distances between hospitals. To do this, we needed the 2010 Census Zip Code dataset from the United States Census Bureau to get geolocation coordinates for every zip code in the United States, which we then used to calculate distances.

We also constructed a Flight network where hospitals are nodes and where the edges capture the amount of air travel between the airports nearby pairs of hospitals. Constructing these edges required not only access to the number of flights being undertaken between all airports in the country but also geolocation data mapping airports to their latitude and longitude coordinates. These data are extracted from the 2009 ASA Data Expo dataset. [26]

Population and other: To create population-level features, we partitioned the US into clusters based on population. Doing this required having population data for each zip code in the US, which is provided in the 2010 Census Zip Code dataset as well. We used publicly available Google Flu Search Trend data as an additional data source in engineering features.

4. Methodology

4.1 Prediction Model

We want to predict whether a hospital in the United States will diagnose at least 1 patient with influenza in a given month given features generated using data from previous months. Influenza is typically transmitted through direct contact or through the air. Many studies suggest that ground and air transportation networks generally provide a good approximation for modeling epidemic spread.

In our study, we constructed three networks based on both empirical data and inferred edges. Our networks try to model the mobilization of people in the US. The empirical networks were based on measures of "distance" -- physical proximity and airport activity -- and the inferred network was based on Rodriguez et al's network inference algorithm. Properties of each of these networks were then used as features in our prediction model. We also cluster hospitals using a similarity metric based on the population of the zip code that each hospital resides in and then use the diagnosis records of all hospitals in a cluster to aid in prediction. Finally, we incorporate non-network features such as the season of the month as well as Google Flu trend data for that month.

In the following section, we will delve into details about our generated networks, the features for our prediction model, the machine learning algorithms, the criteria we use to evaluate the quality of a model's fit to the data, and details regarding hyperparameter selection.

4.1.1 Features

4.1.1.1 Networks Features

4.1.1.1.1 Euclidean Distance Network

The Euclidean distance network measures the physical distance between hospitals. It captures the intuition that hospitals located near other hospitals that have seen influenza cases are more likely to see infections in the future.

Ideally we would have liked to have used distances between hospitals as reported by a service like Google's Distance API to generate this network. However, the usage limits on the service combined with the sheer number of hospitals in our dataset prevented us from doing so.

Instead, we estimated distances based on zip codes. To do so, we mapped each hospital to the zip code to which it belongs. Next, we use United States Census Bureau information to get the geo-location coordinates of that particular zip code in question. Given the geo-location coordinates of two hospitals, we can now easily calculate the distance between them (assuming the United States is a flat plane). We select, for each hospital, the nearest E hospitals, and we generate undirected edges between these E pairs. For each of these E closest hospitals for a given hospital for a given month and a given year, we encode the number of recorded diagnoses of influenza in the previous month at those E hospitals as features.

4.1.1.1.2 Air Transportation Network

These edges are designed to measure the level of air travel activity between two hospitals so as to capture the idea that influenza tends to propagate between areas located near busy airplane hubs. Hospitals close to airports that have many flights between them will spread influenza to each other. For every hospital in our network, we find all airports within a 70 mile radius of it (about an hour of driving distance). For each pair of hospitals we want to determine if they are near different airports that are connected by flights. The weight of this edge is then the number of flights between these two airports. If the hospitals are near multiple airports, then the airport pair is chosen that has the highest level of air travel activity. For each hospital, we generate F undirected edges between the hospital pairs that exhibit the highest level of air travel activity between themselves. As in the Euclidean network, for each of of the F hospitals, we encode as features the number of recorded diagnoses of influenza in the previous month.

4.1.1.1.3.Network Inferred From Disease Cascades

In order to infer the transmission network, we utilize observed rates of infection at a hospital level. More specifically, we infer the structure of the underlying network based on the diffusion of various contagions. Using the HCUP CCS diagnosis code mapping, we identified 4 relevant categories in our dataset: bacterial, viral, influenza and tuberculosis. The diagnosis codes in these categories are mostly airborne diseases and can spread easily. Each category contains many diagnoses codes, see Table 2 for summary.

Category	Tuberculosis	Bacterial infection	Viral Infection	Influenza
Number of ICD9 disease codes in this category	426	107	140	15
Total Number of diagnoses in 2009	586,801	439,729	363,798	30,532
Total Number of diagnoses in 2010	629,627	444,593	385,997	5,614
Total Number of diagnoses in 2011	742,613	487,082	439,352	15,734

Table 2. CCS disease category statistics.

Note that each ICD9 code is composed of a 3-digit disease prefix and then a billable-code suffix. For each 3-digit prefix of the diagnosis codes, and for every year of data, we generate a cascade of hospitals by looking at the earliest time in the calendar year when a hospital reported a diagnosis of the code in question. The year and month is used as the timestamp for the hospital in this cascade. Given multiple cascades, we can infer the underlying network based on Rodriguez and Leskovec's algorithm [9].

In particular, we want to find a network that maximizes the likelihood of the given cascades. Assuming the cascades are independent of each other, we can write this as:

$$\widehat{G} = argmax_G P(C|\ G)\ = argmax_G \prod_{c \in C} P(c|G)$$

where

$\widehat{G}$ is the estimated network
G is the underlying network
C is the set of cascades that are provided as input
c is a single cascade in C

Note that there are some caveats regarding the usage of the inferred network in our prediction model. In particular, we are concerned about using future data to predict past events. It would be a mistake to include diagnosis data from, say, December 2010, in the cascades that produce an inference network which is then used as features for prediction of samples in February 2010. Therefore, for every month in our dataset, we generate a new inference graph built from cascades that use data up to time t-1. When training on a (sample, label) pair $(x_h^{(t)}, y_h^{(t)})$ at time t, we only use features from the inference network generated from cascades utilizing data up to time t-1.

To get a sense of the amount of data involved, the following table summarizes the statistics about the cascades we use for June and December of each year of data we have:

Month, Year	**Number of Cascades Used**	**Average Number of Hospitals (i.e., Events) per Cascade**

June, 2009	55	205
December, 2009	56	260
June, 2010	110	234
December, 2010	112	255
June, 2011	167	240
December, 2011	167	255

Table 3. Cascade statistics.

Given the inference network, we want to use its edges as features in our prediction model. In the Euclidean and air transport network, we simply pick the E and F closest hospitals as features. However, the edges in the inference network are unweighted, so there is no intuitive way to choosing the "best" edges. Therefore, given a hospital h, we want to compute the average cases of influenza diagnosed in its neighboring hospitals in the previous month as a feature in our prediction model.

4.1.1.1.4 Population Zone Clustering

The networks described in 4.1.1.1 to 4.1.1.3 primarily capture pairwise similarities between hospitals. In order to capture community-level influenza activity, we used spectral clustering to cluster zip codes in the United States into zones based on the populations of adjacent zip codes. Formally, given a number of clusters C, we define first compute the Laplacian matrix L as follows:

$$L = D - W$$

Where:

$$D_{ii} = degree(zipcode_i) = \Sigma^n_{j=0} 1\{distance\_in\_miles(zipcode_i, zipcode_j) < 70\}$$

$$W_{ij} = 1\{distance\_in\_miles(zipcode_i, zipcode_j) < 70\} \ (1 / (population(zipcode_i) - population(zipcode_j))^2)$$

$1\{X\}$ is the identity function

We then extract the C generalized eigenvectors corresponding to the lowest C eigenvalues of L. Let $U \in R^{nxc}$ be the matrix where the ith generalized eigenvector of L is the ith column. We then cluster the rows i=1..n of U by running k-means with C centroids and assign $zipcode_i$ to the cluster we assigned the ith row of U.

To extract features from this clustering for use in our prediction, we perform the following calculation. For a given hospital h at time t, we sum up the number of influenza diagnoses at time t - 1 in all hospitals within the same cluster as h and use that as a feature for h.

4.1.1.2 Non-Network Features

The literature tells us that influenza is more likely to occur during certain months of a year [25]. Therefore, we added three variables to account for seasonality: whether the current month is a spring month, a summer month, or a autumn month. There has also been extensive research correlating Google's National Flu Index data to the number of influenza diagnoses in the country at large [3]. So, for any given month and year, we average the weekly Google Flu Index values for the previous month and use it as a feature for all hospitals for this month. Note that these features are very general and, actually, identical for all hospitals at a given time t: this is because the focus of our project is on the use of network features to aid classification.

4.1.1.3 Feature Vector for Classification

Each sample in our dataset is denoted as $x_h^{(t)}$ where h is a particular hospital and t corresponds to a particular time, which is a particular month m and year y combination. $x_h^{(t)}$ is composed of the following E + F + 1 + 6

features. We have E and F features from the Euclidean and Flight network, 1 feature from the inferred network and the remaining 6 features, which are: the Google Flu Trends index, the season of m, and the number of influenza diagnoses last month in hospitals within the same population zone as h.

$x_h^{(t)}{}_{1...E}$	the number of influenza diagnoses during the previous month in the ith most physically nearby hospital to h, i=1..E
$x_h^{(t)}{}_{E+1...E+F}$	the number of influenza diagnoses during the previous month in the hospital that exhibits the ith highest air travel activity between h and any other hospital, i=1..F
$x_h^{(t)}{}_{E+F+1}$	the average number of influenza diagnoses during the previous month in hospitals which are the source node for an edge with h as the destination node
$x_h^{(t)}{}_{E+F+2}$	the total number of influenza diagnoses during the previous month in all hospitals located in the same population zone as h, excluding h itself
$x_h^{(t)}{}_{E+F+3}$	the averaged Google Flu Trend index for the previous month
$x_h^{(t)}{}_{E+F+4}$	1 if month m is a spring month 0 otherwise
$x_h^{(t)}{}_{E+F+5}$	1 if month m is a summer month 0 otherwise
$x_h^{(t)}{}_{E+F+6}$	1 if month m is an autumn month 0 otherwise
$x_h^{(t)}{}_{E+F+7}$	1 (i.e., a bias term)

The response $y_h^{(t)}$ for a given sample $x_h^{(t)}$ is then:

$y_h^{(t)}$	1 if h had at least 1 diagnosis of influenza for month m of year y 0 otherwise

4.1.1.4 Classifiers

4.1.1.4.1 Logistic Regression

We experimented with several machine learning classifiers such as C-SVMs with linear and nonlinear kernels, Random Forests, and various Naive Bayes models. We found that Logistic Regression - a very simple, robust classifier - delivered the best results on our validation data. We use liblinear's L2-regularized Logistic Regression implementation for all of our work. As described by the library's authors [13], liblinear attempts to solve the following primal optimization problem:

$$min_w\, w^T w/2 \; + \; C\Sigma_{i=1}^{n}\, log(1 + exp(-y^{(i)}\, w^T x^{(i)}))$$

Where i=1..n are the training samples and C is a regularization parameter. As the class label distributions in the training set are frequently not equal, we adjust the model's cost of misclassifying a label by the inverse of its proportion in the training dataset.

4.1.1.4.2 Support Vector Machines

We frequently use the C-SVM as a secondary model to highlight how much better Logistic Regression does at predicting our dataset. We rely on libsvm's implementation of a C-SVM along with a radial basis function (RBF) kernel to perform classification. As with the Logistic Regression model, we adjust the SVM's cost of misclassifying

a label by the inverse of its proportion in the training dataset. According to the libsvm documentation [25], we are solving the following optimization problem:

$$min_{\alpha}(1/2)\alpha^T Q\alpha - e^T a$$
$$subject\ to\ y^T\alpha = 0,\ 0 \le \alpha \le C,\ i = 1..n$$

Where:

$$e = [1, ..., 1]^T \in \Re^n$$
$$Q_{ij} = y^{(i)}y^{(j)}exp(-\gamma||x^{(i)} - x^{(j)}||^2)$$

And note again that i,j = 1...n, where n is the number of samples and C is a regularization parameter.

4.1.2 Evaluation Method and Hyperparameter Selection

For each sample in our dataset, which encodes information about a hospital given information up until the previous month, we try to predict whether this hospital will admit at least 1 patient diagnosed with influenza in this month. Hence, we are dealing with a binary classification problem.

We partitioned the dataset into three sections: training, validation, and testing segments. To prevent using future data to predict past events, we partitioned our dataset by time. All samples encoding information about a hospital during the year 2009 are used for training. Our validation set consists of all samples from 2010, and our testing set consists of all samples from 2011. We partition by year because the literature spends great effort describing how seasonality is a very important indicator in predicting influenza outbreaks. Therefore, if we only included samples from a few seasons of a given year in the validation or testing data sets, the distribution of class labels across the partitions would be very skewed, resulting in suboptimal classification performance.

We use the training data to fit our classifier, and we use the precision and recall metrics (specifically, F-2 scores as we care far more about false negatives than false positives) from its performance on the validation set to tune hyperparameters to achieve even higher classification performance.

Another important consideration is to ensure that the model does not overfit on specific hospitals in the training set. In particular, we are concerned that the given features for a hospital might uniquely identity the hospital with regards to the classifier. This is problematic because this would suggest that our classifier is unable to generalize to hospitals that are not presented in the training data, which we feel is a very important property that we want our classifier to have. However, excluding hospitals that are present in the testing data from the training data would decrease the size of our training set when we are already suffering from small data set sizes due to the amount of missing data (the missing data issue is discussed in more detail in 4.1.3).

In order to verify that our classifier is generalizing as well as we'd like, we report our classifier's performance on two different tests: one where we filter out all hospitals that are present in the non-training set partition from the training data, and one where we do not. That is, if a hospital is present in the non-training data, then in this latter model it will be as if that hospital never existed prior to January 2010 in the case of testing on validation data or prior to January 2011 in the case of testing on testing data. This also means that the filtered-out hospitals are not used as features for any hospitals that are still left in the training data. If generalization performance on these two models is similar, then we know that the hospital removal was too conservative of an approach.

Initially, you would think there are three hyperparameters that we need to choose: the network edge parameters E,F, and also C, which determines how many population zones to create for spectral clustering. However, it is also an open question whether we should even use a certain network for our classification task; prediction might work better if we simply disregard all information from one of the networks we created. So we also include the question of whether to use a particular network in our hyperparameter selection process. This manifests itself by seeing what the F-2 scores are like on the validation set when E = 0, F = 0, or when we don't include the feature from the Inferred network.

We used the F-2 score obtained from testing on the validation set to select these hyper parameters using a greedy grid-search approach where we use or don't use data from the Inference network, $E \in \{0, 1, .., 5\}$, $F \in \{0, ..., 3\}$ and C

$\in$ {25, 50, 75, 100, 125, 150, 175, 200}. We selected these hyperparameter windows after a lot of manual trial and error which gave us a good intuition of the general range of hyperparameter values our classifiers do best with. Once we have found our optimal hyperparameters, we train our model using the training and validation data, and test on the withheld testing data. We then output our results on the training+validation and test sets in the Results section.

4.1.3 Dealing with Missing Data

Due to confidentiality laws in some states such as Texas, the location and zip codes of hospital in those states are not listed. The address and zip codes are important because all our networks are geospatial. Without the address or the zipcode we cannot use any data originating from these hospitals. It turns out that 40% of the hospitals in the dataset are not assigned a zip code (see Tables 1 & 4 for a comparison).

Due to the large amount of missing data, different strategies of dealing with missing data were considered. One strategy we considered was the following: for each missing value of a particular feature, assign the mean value for this feature seen in the training set. However, this didn't work well: the additional data caused our classifiers to achieve very low F-2 scores. We found that simply removing all samples with missing data yields the best classifier performance. That is, when building feature vectors for our models, if we are using specific parameters for E and F, and we find that sample $x_h^{(t)}$ does not have enough information to create E edges in the Euclidean network (which can happen if the hospital has no zip code registered) or F edges in the Flight network (which occurs when a hospital is next to a very isolated airport or none at all), then this sample is simply thrown away.

We thus focus on the subset of hospitals whose zip codes are provided. Below is a summary of the subset of hospitals in the NIS dataset that are usable:

Year	**2009**	**2010**	**2011**
How many total hospitals in the NIS data that includes zip codes	575	595	601
How many hospitals with zip codes reported cases of influenza	485	367	467
Zip Codes (2010 Census)	33,120	33,120	33,120

Table 4. Statistics regarding missing data in NIS dataset.

4.2 Baseline Models

To evaluate our model, we want to compare its performance against two naive baseline models. Specifically, we want to compare our results with baseline models that use simple, non-network features so that we can evaluate the predictive power of our prediction model. We considered the following models:

4.2.1 Always Predict Majority Label Model

The most naive model was one that completely ignores all features and simply outputs the majority class label of the training set. Given a training set of n samples, the majority label is defined as:

$$majoritylabel = argmax_j \Sigma_{i=0}^{n} 1\{y^{(i)} = j\}$$

4.2.2 Use Only the Previous Month Model

The next model to implement was, for a given hospital h at time t, to predict whether h will have any diagnoses of influenza at t, we simply use one feature: whether h had any cases of influenza diagnosed at time t-1. Using the notation that we defined in 4.1.3 for labels, our model is therefore:

$$y'^{(t)}_h = y_h^{(t-1)}$$

Where $y'_h{}^{(t)}$ denotes our estimated label for this hospital at time t, and $y_h{}^{(t-1)}$ denotes that actual ground truth label designating whether h had any cases of influenza at time t-1.

5. Results and Discussions

5.1 Networks

As we have mentioned previously in 4.1.1.1 to 4.1.1.3, we generated 3 networks to capture the various potential paths of infection in the United States. The nodes represents hospitals in our dataset while the edges capture the different paths by which disease propagates over the network. To get a better intuition about the properties of these networks, we included examples of several networks we generated. We overlaid these network on top of a map of the United States to better illustrate how hospitals are connected in each network.

5.1.1 Euclidean Distance Network

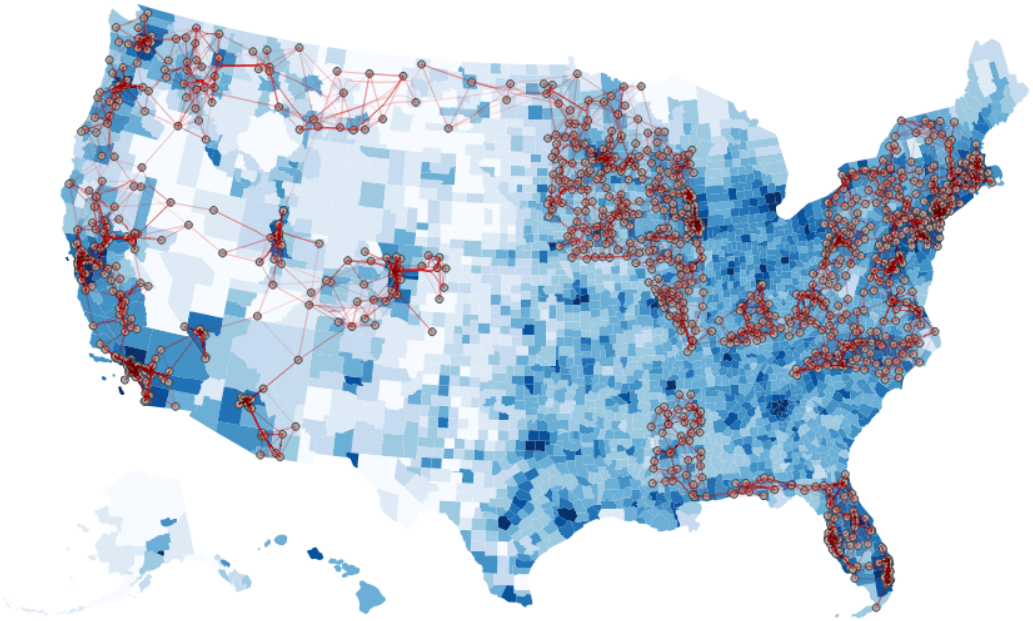

Figure 1. Euclidean Distance Network with 5 edges per hospital (E=5)

5.1.2 Flight Network

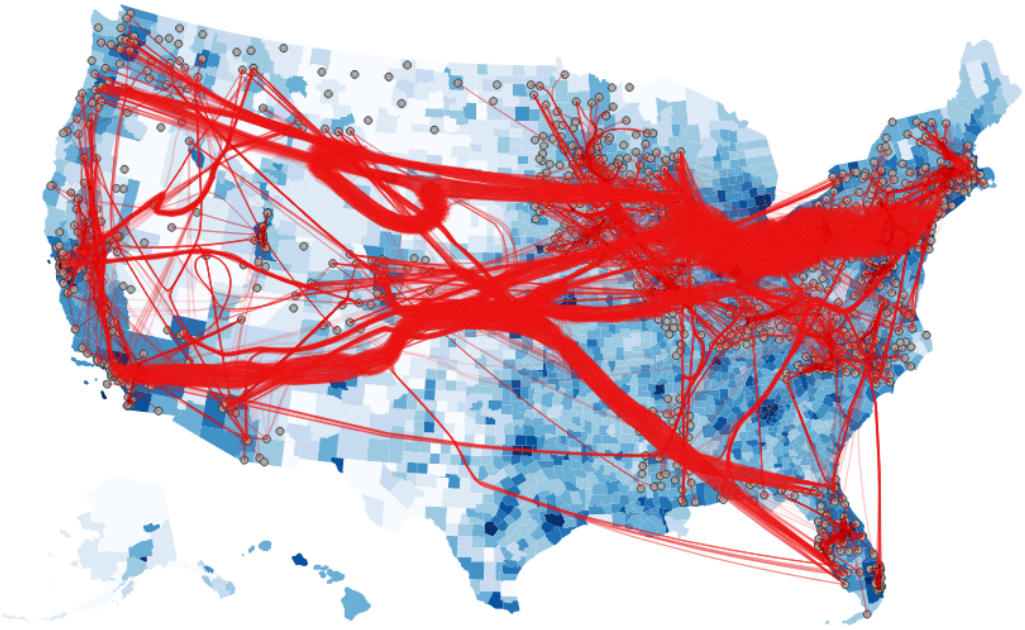

Figure 2. Flight network with 5 edges per hospital (F=5)

5.1.3 Network Inferred From Disease Cascades

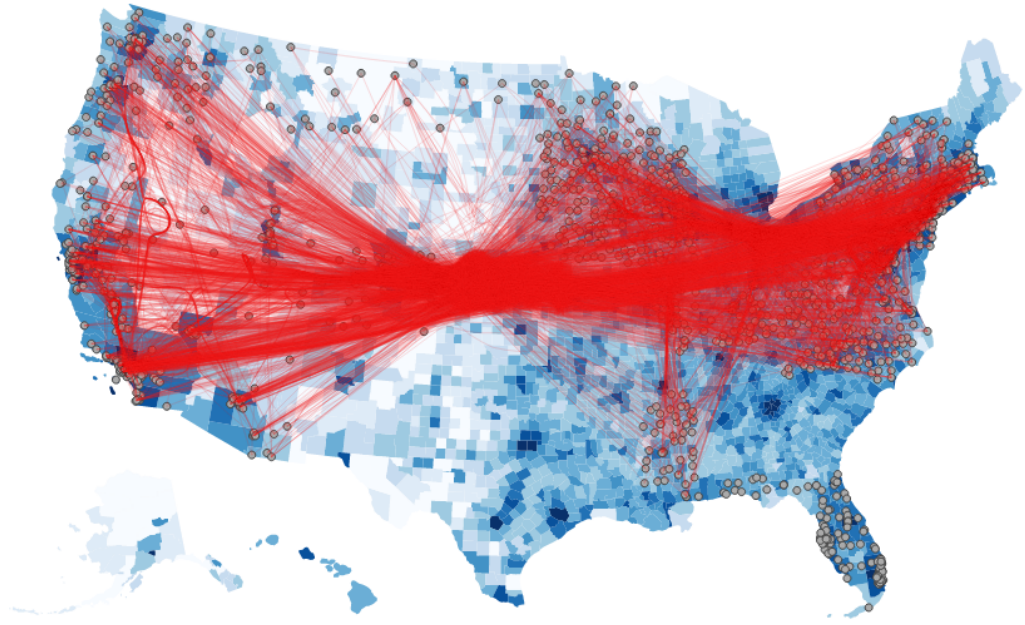

Figure 3. Inferred network from disease cascades built using diagnosis data prior to December, 2011

Network	Clustering Coefficient	Mean Degree	Diameter
Euclidean Distance Network	0.559	7.88	49
Flight Network	0.138	10.62	9

Network Inferred From Disease Cascades	0.038	10.13	8

Table 5. Statistics of network models.

The Euclidean distance network is constructed by adding edges to the top E neighbors of a hospital. The intuition is that diseases spread to nearby regions. From the network visualization, we see that the Euclidean distance network has many short edges and nodes tend form clusters with nearby nodes. As a result, it has the highest clustering coefficient. At the same time, because of the lack of long edges, it has a high diameter which means that it will not capture spread of disease over long distance. To compensate for the lack of long distance edges, we added the flight network which captures spread of a contagion from cities to cities. The flight network has many long edges and it has a much smaller diameter than the Euclidean distance network. The network inferred from disease cascade has mostly long edges and also the lowest clustering coefficient. We should note that it is surprising to us that the inferred network consists mostly of long edges: we were expecting a mix of both long and short edges as diseases spread to nearby regions and also from cities to cities via air transportation networks.

5.2 Prediction Performance & Analysis

The goal of our project is to predict whether a hospital in the United States will diagnose at least 1 patient with influenza in a given month given features generated using data from previous months. In this section, we will compare the performance of our model specified in 4.1 with the two baseline models detailed in 4.2.

5.2.1 Baseline Models

5.2.1.1 Always Predict Majority Label Model

In our training+validation set, there are 13,465 samples that have the 'Not Infected' label, and 7,212 instances with the 'Infected' label. Therefore, 'Not Infected' is the majority label, and we assign the 'Not Infected' as our prediction for all samples in the training, validation, and testing sets. The results are summarized in the following table:

Dataset	Accuracy	Not Infected Accuracy	Infected Accuracy	Precision	Recall	F-2 Score
Training + Validation	0.6960	1.0	0	0	0	N/A
Testing	0.7428	1.0	0	0	0	N/A

Table 6. Majority Label Model Results.

Note that precision and recall are both 0 because we consistently output 'Not Infected' and therefore the number of "true positive" is 0. Furthermore, we couldn't calculate the F-2 score since Precision and Recall are 0. It is surprising that 'Not Infected' is the majority label. This implies that most samples do not even report a single case of influenza in our dataset.

5.2.1.2 Use Only the Previous Month Model

We again trained on the training and validation sets and tested our model using the testing set. The following are the results we get when we use two classifiers:

Classifier	Dataset	Accuracy	Not Infected Accuracy	Infected Accuracy	Precision	Recall	F-2 Score
Logistic Regression	Training + Validation	0.7884	0.8637	0.6159	0.6638	0.6159	0.6249
Logistic Regression	Testing	0.7958	0.8727	0.5736	0.6094	0.5736	0.5804
SVM (rbf kernel)	Training + Validation	0.7884	0.8637	0.6159	0.6638	0.6159	0.6249

SVM (rbf kernel)	Testing	0.7958	0.8727	0.5736	0.6094	0.5736	0.5804

Table 7. Previous Month Model Results.

Note that all the performance metrics have the same value for both Logistic Regression and the C-SVM. Recall that we only use one feature in this model: whether h had any cases of influenza in the previous month. As a result, there are only 4 possible unique combinations of data points in our dataset for this model:

- $y_h^{(t-1)} = 1$, $y_h^{(t)} = 1$
- $y_h^{(t-1)} = 1$, $y_h^{(t)} = 0$
- $y_h^{(t-1)} = 0$, $y_h^{(t)} = 1$
- $y_h^{(t-1)} = 0$, $y_h^{(t)} = 0$

Thus, we don't expect to see very much change (in this case, none at all) when shifting from a robust classifier like Logistic Regression to a high-variance one like a C-SVM with a RBF kernel.

The results show that the model does a decent job of accurately predicting the labels for the training+validation data: accuracy is rather high, but the model does a better job of predicting the 'Not Infected' label, judging by its accuracy on that label. Not only is the F-2 score decently high, but note that the Precision and Recall scores are quite close to each other, and so the model isn't producing false positives at a rate that's much higher than its false negative rate or vice versa. Most importantly, the model generalizes very well: the F-2 scores on the testing data are very close to the scores on the training+validation data.

5.2.2 Prediction Models

We present the results from our Prediction model below. Note that, as discussed in 4.1.2, we are concerned about the generalization of our model on hospitals not present in the training set. Therefore, we are providing two sets of results: one for the prediction model run where we remove hospitals present in the testing set from the training set (5.2.2.2), and one where we did not exclude any hospitals from the training set (5.2.2.1).

5.2.2.1 Testing Hospitals Not Filtered from Training Set

We first present our results when we train on the training+validation set and test on testing set. Included are also the hyperparameters we found with grid search for this specific classifier by following the methodology outlined in 4.1.2. For conciseness let I = 1 indicate that the grid search found we should use the Inference network, and let I = 0 indicate otherwise.

Classifier	Dataset	Optimal Hyperparameters	Accuracy	Not Infected Accuracy	Infected Accuracy	Precision	Recall	F-2 Score
Logistic Regression	Training + Validation	E=1, F=2, C=100, I=0	0.6910	0.8442	0.5282	0.7614	0.5282	0.5627
Logistic Regression	Testing	E=1, F=2, C=100, I=0	0.8216	0.9443	0.5181	0.7898	0.5181	0.5564
SVM (RBF kernel)	Training + Validation	E=5, F=3, C=100, I=0	0.8361	0.8555	0.81723	0.8532	0.8172	0.8242
SVM (RBF kernel)	Testing	E=5, F=3, C=100, I=0	0.2741	0	1.0	0.2741	1.0	0.6538

Table 8.Testing Hospitals Not Filtered from Training Set Model Results.

First, note that, overall, Logistic Regression fit the training+validation data relatively well: the total accuracy is relatively high, and its F-2 score is also high. Note, however, that the precision score is far higher than the recall score: it looks like the classifier is being too conservative in its classification of the 'Infected' label, to the detriment of its F-2 score. However, it looks like the classifier maintained a very similar F-2 score on the testing set: this suggests that it generalized very well to the testing set. The noticeably higher accuracy we get on the testing set is an artifact of the fact that the proportion of class labels in the testing set is slightly different from that in the

training+validation set and from the fact that the exclusion of those samples with insufficient data has slightly altered the label distributions. Unfortunately, note that the model achieves a lower F-2 score on both training+validation and on the testing sets than the Previous Month model.

The C-SVM fits the training+validation data extremely well: the F-2 score is extremely high, and the Precision and Recall metrics are very close to each other. In fact, it fit the data far too well: the classifier only outputs a prediction of 'Infected' on the testing set! Despite all our attempts to regularize its learning by tuning its internal regularization parameter, we could only force the classifier to solely predict either 'Not Infected' or 'Infected' for the entire validation set, and we thus chose to predict the 'Infected' label. Another remedy to this overfitting is to train the classifier on a larger dataset and is thus not a realizable strategy. In the end, the C-SVM achieves a slightly higher F-2 score on the testing data than the Previous Month Model, but it sacrifices all accuracy on the 'Not Infected' label.

The following chart shows the effect of varying individual parameters on the performance metrics of the Logistic Regression model. As we can't show the entire grid search space, we fix the parameters at their optimal values as found through grid search, and we then show the performance metrics by varying one parameter over its entire search window. We omitted showing the hyperparameter search results for the C-SVM due to that model's mediocre classification performance.

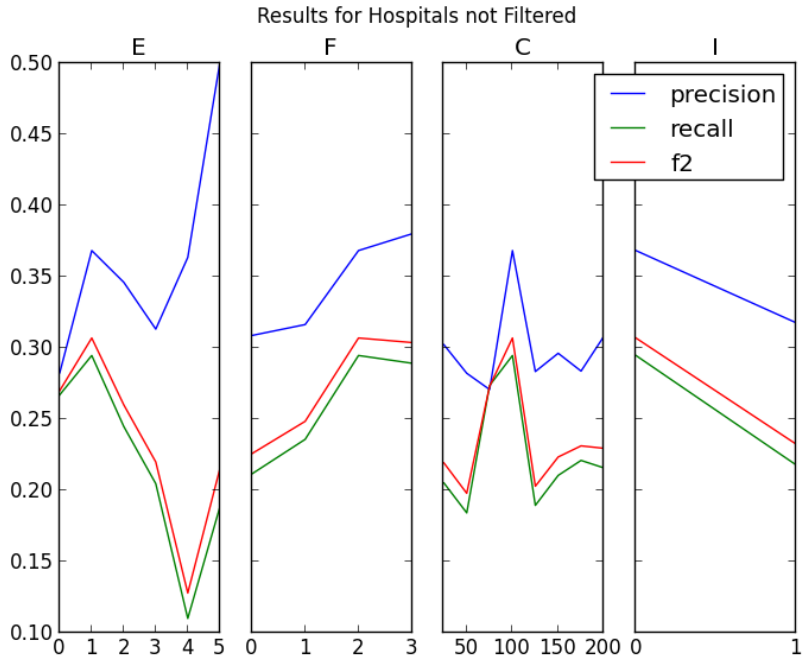

Figure 4. Precision, recall and f2 scores of validation set as we vary the individual hyperparameter.

Recall scores typically fall as we add additional Euclidean edges while precision scores rise. As a result the grid search choose E=1. This is not true for edges from the Flight network, though: those in general seem to aid in improving F-2 scores (until F=3 where we see a drop in performance). There is a sharp peak in the graph for the C parameter that made finding the optimal value unambiguous. Finally, it is extremely clear that including features from the Inferred network hinders classification performance on the validation set: both Recall and Precision scores drop dramatically with the inclusion of features from this network.

5.2.2.2 Testing Hospitals Filtered from Training Set

Again, we present our findings once we have found the optimal hyperparameters per classifier and then used those values to train on the training and validation sets and then test on the testing set:

Classifier	Dataset	Optimal Hyperparameters	Accuracy	Not Infected Accuracy	Infected Accuracy	Precision	Recall	F-2 Score
Logistic Regression	Training + Validation	E=2, F=3, I=0, C=175	0.6886	0.8179	0.5576	0.7513	0.5576	0.5879
Logistic Regression	Testing	E=2, F=3, I=0, C=175	0.7908	0.9130	0.4670	0.6697	0.4669	0.4970
SVM (RBF kernel)	Training + Validation	E=5, F=3, I=0, C=175	0.8979	0.9014	0.8948	0.9119	0.8948	0.8982
SVM (RBF kernel)	Testing	E=5, F=3, I=0, C=175	0.2960	0	1	0.2960	1	0.6776

Table 9.Testing Hospitals Filtered from Training Set Model Results.

As we can see, Logistic Regression fits the training+validation data in this model about as well as it fit the training+validation data in 5.2.2.1: the F-2 scores are about the same, and you again see that Precision is again about 20% higher than Recall. The testing data results also aren't too different from the results of the previous model: the F-2 score achieved is only a bit lower, and Recall is a bit lower than in the previous model as well. Note, however, that the Precision score is quite a bit lower than in the previous model. This might be attributed to the fact that the model was identifying specific hospitals and outputting the correct predictions. However, as the F-2 score calculation downplays the value of Precision as compared to Recall, performance on this metric isn't greatly affected.

Thus, there is some evidence that overfitting is occurring on the training hospitals, but taking care of the issue doesn't lead to a large change in F-2 scores, our key metric. So, it seems like filtering the hospitals from the training set was probably too conservative a move. Regardless, the model fares worse from the perspective of F-2 scores than the Previous Month model.

Unfortunately, we see that the C-SVM, despite our best attempts, also overfits the training+validation data in this model as well. However, in general, it achieves higher performance on both datasets: its F-2 scores are noticeably higher than in the 5.2.2.1 model. It goes without saying that we'd still prefer the Previous Month model to this one.

As in 5.2.2.1, we present the results of the hyperparameter grid search for Logistic Regression on this dataset below.

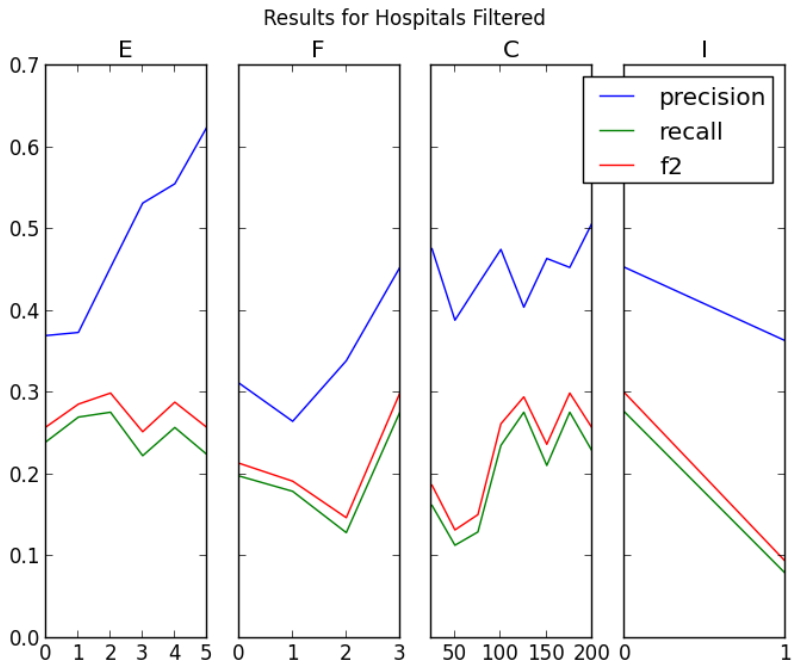

Figure 5. Precision, recall and f2 scores of validation set as we vary the individual hyperparameter.

As in the previous model, adding more edges in the Euclidean network seems to, in general, adversely affect Recall scores while improving Precision scores. Interestingly, generalization performance suffers if we use F=1 but in general rises as we increase F's value, which was also the finding from 5.2.2.1. The nice peak from the C graph for 5.2.2.1 is gone, and we see that in general, generalization performance does better as you increase the number of clusters. Finally, just as in the previous model, it is extremely clear that features from the Inferred network should not be used: both Recall and Precision scores fall dramatically when we harvest that network for features.

5.2.2.3 Testing Hospitals Not Filtered from Training Set + Previous Month Model

As we failed to beat the baseline, we were wondering how our model's performance would change if we added the Previous Month model's features to it. Since we established that filtering out hospitals from the training data didn't alter our metrics very much, we only decided to perform this experiment on the model where we did not filter out hospitals from the training set. We once again chose the best hyperparameters by training on the training data and testing on the validation data, and we then collected final metrics by testing on the test partition. These results are described below:

Classifier	Dataset	Optimal Hyperparameters	Accuracy	Not Infected Accuracy	Infected Accuracy	Precision	Recall	F-2 Score
Logistic Regression	Training + Validation	E=5, F=3, I=0, C=50	0.7241	0.7251	0.7233	0.7523	0.7233	0.7289
Logistic Regression	Testing	E=5, F=3, I=0, C=50	0.8117	0.9083	0.5818	0.7273	0.5818	0.6061
SVM (RBF kernel)	Training + Validation	E=5, F=3, I=0, C=25	0.8969	0.9212	0.8758	0.9277	0.8758	0.8857
SVM (RBF kernel)	Testing	E=5, F=3, I=0, C=25	0.2960	0	1	0.2960	1	0.6776

Table 10.Testing Hospitals Not Filtered from Training Set + Previous Month Model Results.

Firstly, we see that Logistic Regression fits the training+validation data extremely well: its Precision and Recall scores are very close to each other, and even the 'Not Infected' and total Accuracy metrics are close to the F-2 score. This is in constrast to the model in 5.2.2.1 which had a far better Precision than Recall score. In terms of generalization, we see something fantastic: the F-2 score is higher than it is in the Previous Month model! Indeed, it looks like this model retains a gap between Precision and Recall on the testing set that we saw in the 5.2.2.1 model, but the gap has shrunk enough so that the model's F-2 metric beats that of the baseline. So, we finally have a model that beats the Previous Month model, but it has to include that model's features to do it.

Expectedly, the inclusion of the additional feature wasn't enough to change the behavior of the C-SVM on our datasets: it still only predicts the 'Infected' label on the testing data.

Finally, the results of our hyperparameter search for Logistic Regression are presented below:

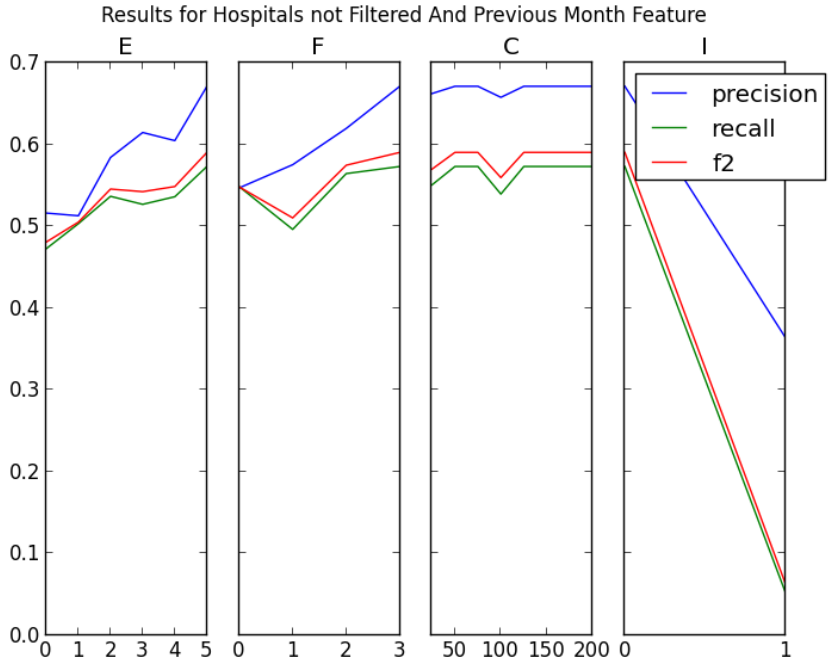

Figure 6. Precision, recall and f2 scores of validation set as we vary the individual hyperparameter.

In contrast to the previous two models, adding more edges from the Euclidean and Flight network improve precision, recall and F-2. However, varying C, the hyperparameter for spectral clustering, has little effect on classification performance. The inclusion of the previous month feature (and the new choices for the other hyperparameters) has resulted in the classifier consistently placing less emphasis on the clustering feature than it did in previous models. However, just as in the other two models, including features from the Inferred network results in worse generalization performance, both in terms of Precision and Recall.

5.3 Project Challenges

5.3.1 Data

It is clear to us that the NIS dataset, which is constructed of administrative data, is not an ideal dataset to model disease propagation. Although each year of data typically includes 8 million records from over a thousand hospitals, as we mention in 4.1.3, states have different disclosure rules, and many records are incomplete. For example, almost 40% of those hospitals are in states that suppress disclosing any information about the location of the hospital; thus, the data originating from those hospitals is effectively useless as we can't tie it to particular subregions of the state. This is a particularly acute problem in the case of Texas, which contains 10% of the hospitals in any given year, is responsible for 9% of the inpatient data, but which doesn't require any location disclosure. Some states like Florida suppress releasing crucial portions like a patient's admission month, and so all records from such states are also useless to us within the context of the current model formulation. We've also found that the same hospitals will get different NIS identification numbers due to administrative changes.

Finally, note that influenza is only contagious for a very small window, and the vast majority of those hospitalized will be the elderly, the young, and the immunocompromised. By the time a person feels ill enough to become hospitalized for flu-like symptoms, they might actually no longer be contagious and only be suffering from secondary diseases such as pneumonia. Thus, hospital records will significantly underestimate the true spread of influenza in the general population. Note that this is the same bias that most of the literature is even more subject to as the basis for much of the research surrounding influenza is CDC influenza mortality data.

5.3.2 Transportation Networks

We use a static representation of the United States air traffic network for all three years of data. This is a rough approximation for the true air traffic network, which changes every day based on a multitude of factors including, importantly, rates of infections of the population living nearby airports. We thus lose out on an extremely important dynamic that could greatly aid our prediction task.

As we've mentioned in 4.1.1.1.1, the Euclidean network assumes the United States is a flat plane and uses the zip codes that hospitals reside in to calculate the distance between all pairs of them. This is also a rough approximation as the United States is most certainly not a flat plane but is extremely diverse in its geography and altitudes. Furthermore, if use consider actual roadways, then it is clear that the actual shortest distance between two hospitals together may deviate rather significantly from their distance within the Euclidean network. Thus, a proper Euclidean network would use the exact distances between hospitals as measured by roadways and would take geography into account during distance calculation.

5.3.3 Inference Network

Another key limitation is that we're missing the ground truth network regarding influenza propagation within the hospital network. Although we compensate for this by leveraging a number of different cascades related to infectious diseases that are fed into the inference engine, our only method of seeing whether the edges are useful is to wait to see changes in the metrics from the classifier's performance.

It should be noted that, as shown in the results in 5.2, the grid search for all of our Prediction models told us to exclude using features from the inference networks. There are a variety of interpretations to this. The most likely one, we feel, is that we didn't have enough cascades to build a robust inference network: note that Rodriguez et al used thousands of cascades to create the inference networks in their work whereas we had less than 200.

6. Future Work

Our prediction system involves many complicated components - transportation network estimation, network inference, spectral clustering, and machine learning classification - in order to track the spread of influenza within the United States. Each of these components uses some level of approximation; in hindsight, we are afraid that these approximations accumulate errors within the pipeline and result in suboptimal prediction performance. There are several areas we could explore to extend our current work:

- The primary purpose of the HCUP dataset is to capture cost utilization information. It might not be suitable for the purpose of predicting influenza propagation. Currently, the hospital victims for influenza are overwhelmingly elderly and most occurrence of influenza is probably not hospitalized. Perhaps a better dataset for capturing the spread of influenza is the Google Flu Trend data broken down by cities. Unfortunately, this dataset is just a time series and doesn't by itself capture cases of influenza.
- Adjust our prediction model to more accurately reflect spread of an epidemic. In our current model, a hospital is considered infected if a single patient is reported as having the disease. Clearly, this generates more false positives in terms of epidemic prediction. To improve, we need to provide more accurate thresholds for when a hospital is "infected".
- Spend the resources and time required to build networks that more closely captures the true ground, flight, and inference networks between hospitals.
- We could include more advanced network features in the feature vector of each sample of the Prediction model (e.g., the number of neighboring hospitals that are also connected to each other).

7. Conclusion

We built two ground and air transportation networks to model the mobilization and thereby the propagation of diseases in the United States. We also constructed an inferred disease propagation network over the United States using multiple diseases cascades we extracted from the HCUP dataset. Furthermore, we used spectral clustering to cluster zipcodes in the United States into population zones. Combining these networks, clusters and other non-network features, we constructed a prediction model using logistic regression to predict whether a hospital in

the United States will diagnose at least 1 patient with influenza in a given month given features generated using data from previous months.

In the end, our Prediction model achieved higher F-2 scores than the Majority Label baseline model, and it only beat the Previous Month baseline model when we incorporated the features from this model into the Prediction model. We also found that even though there was some evidence of overfitting on the hospitals in the training set for the Prediction model, there was not much of a difference in terms of F-2 scores between the Prediction models where we filtered out testing set hospitals from the training set and where we did not, which lead us to conclude that the filtering step was not necessary.

8. References

1. World Health Organization. Influenza fact sheet. http://www.who.int/ mediacentre/factsheets/2003/fs211/en/ (2003).

2. http://www.cdc.gov/flu/weekly/fluactivitysurv.htm.

3. Ortiz et al. Monitoring Influenza Activity in the United States: A Comparison of Traditional Surveillance Systems with Google Flu Trends. PLoS ONE, April 2011, Volume 6, Issue 4.

4. Brammer et al. Seasonal and pandemic influenza surveillance considerations for constructing multi-component systems. Influenza and other Respiratory Viruses 3(2), 5158.

5. Machens et al. An infectious disease model on empirical networks of human contact: bridging the gap between dynamic network data and contact matrices. BMC Infectious Diseases 2013, 13:185.

6. Lipsitch et al. Managing and Reducing Uncertainty in an Emerging Influenza Pandemic. NEJM 361;2.

7. Colizzi et al. Prediction and predictability of global epidemics: the role of the airline transportation network. Proc. Natl. Acad. Sci. USA 103, 2015 (2006).

8. Welch et al. Statistical inference to advance network models in epidemiology. Epidemics. 2011 March ; 3(1): 3845.

9. Gomez-Rodriguez, M., Leskovec, J., and Krause, A. 2012. Inferring networks of diffusion and influence. ACM Trans. Knowl. Discov. Data 5, 4, Article 21 (February 2012), 37 pages.

10. Goldenberg, A.; Zheng, A.; Fienberg, S.; Airoldi, E. A survey of statistical network models. Foundations and Trends in Machine Learning. 2009.

11. Ginsberg et al. Detecting influenza epidemics using search engine query data. Nature, 457,19, February 2009.

12. Manuel Gomez Rodriguez , Jure Leskovec , Andreas Krause, Inferring networks of diffusion and influence, Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, July 25-28, 2010, Washington, DC, USA

13. Fan, Rong-En, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. "LIBLINEAR -- A Library for Large Linear Classification." *LIBLINEAR -- A Library for Large Linear Classification*. National Taiwan University, n.d. Web. 27 Nov. 2013

14. Shi, Jianbo, and Jitendra Malik. "Normalized Cuts and Image Segmentation."*Normalized Cuts and Image Segmentation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Aug. 2000. Web.

15. Von Luxburg, Ulrike. "A Tutorial on Spectral Clustering." *A Tutorial on Spectral Clustering*. Max Planck Institute for Biological Cybernetics, 17 Apr. 2007. Web.

16. D. Holten and J. J. van Wijk., Force-directed edge bundling for graph visualization. *Computer Graphics Forum (Proc. EuroVis 09),* 28 (3): 983-990, 2009.

17. Thompson WW, Comanor L, Shay DK (2006) Epidemiology of seasonal influenza: use of surveillance data and statistical models to estimate the burden of disease. J Infect Dis 194: supp l2S82–S91. doi: 10.1086/507558.

18. Colizza V, Barrat A, Barthélemy M, Vespignani A (2006) The role of the airline transportation network in the prediction and predictability of global epidemics. Proc Natl Acad Sci U S A 103: 2015–2020. doi: 10.1073/pnas.0510525103.

19. Welch D, Bansal S, Hunter DR (2011) Statistical inference to advance network models in epidemiology. Epidemics 3: 38–45. doi: 10.1016/j.epidem.2011.01.002.

20. Ginsberg J, Mohebbi MH, Patel RS, Brammer L, Smolinski MS, Brilliant L. Detecting influenza epidemics using search engine query data. Nature 2009;457:1012-1014

21. Crépey P, Barthélemy M. Detecting robust patterns in the spread of epidemics: a case study of influenza in the United States and France. Am J Epidemiol. 2007; 166:1244.-51

22. Keller, Nat. "Math Forum - Ask Dr. Math." *Math Forum - Ask Dr. Math*. N.p., n.d. Web. 13 Nov. 2013.

23. Rodriquez, Manuel, Jure Leskovec, and Bernard Schölkopf. "Algorithm: INFOPATH."*Structure and Dynamics of Information Pathways in On-line Media*. N.p., n.d. Web. 13 Nov. 2013

24. Shaman, J. and Karspeck, A. (2012). Forecasting seasonal outbreaks of influenza. Proceedings of the National Academy of Science http://www.pnas.org/content/early/2012/11/21/1208772109

25. Chang, Chih-Chung, and Chih-Jen Lin. "LIBSVM: A Library for Support Vector Machines." (n.d.): n. pag. *LIBSVM: A Library for Support Vector Machines*. National Taiwan University, 2001. Web

26. http://stat-computing.org/dataexpo/2009/