DENGAI: PREDICTING DISEASE SPREAD

PROJECT FINAL REPORT

Juan Langlois

December 7, 2021

1 Introduction

The project's goal is to build a model that can forecast the spread of dengue infections in Central America. In particular, I will use the data of the DrivenData competition DengAI: Predicting Disease Spread. Dengue fever is a mosquito-borne disease that occurs in tropical and sub-tropical parts of the world. Because mosquitoes carry the disease, the transmission dynamics are related to climate variables such as temperature and precipitation. The task is to predict the number of dengue cases each week and in each location based on environmental variables describing changes in temperature, precipitation, vegetation, and more (DrivenData [2022]).

Because epidemic growth slows as the population of susceptible individuals is exhausted. This system feedback can lead correlations between variables to switch signs during different periods, resulting in a net correlation of zero. Previous prediction models of dengue outbreaks used phenomenological and mechanistic equation-based approaches. However, these models may not fully capture interdependence between climate and susceptible availability. Nova et al. [2021] proposes as an alternative the use of empirical dynamic modeling (EDM) to identify and model mechanisms driving dengue epidemics. EDM is based on reconstructing the system dynamics evident in time series, without assuming fixed relationships. Relationships among variables can change through time if interactions among variables are context-dependent.

The model proposed by Nova et al. [2021] improves forecast skill over recent, state-of-the-art models for dengue incidence. Inspired by these results, we will experiment with Hidden Markov Models (HMMs) and their potential to predict dengue incidence. In particular, we will test three different architectures: (i) Poisson HMM, (ii) Poisson Generalized Linear Model (GLM) HMM, and (iii) Poisson Input-Output HMM. Each model adds a level of complexity that should allow more flexibility and flow of information.

The first model is the basic HMM architecture with constant transition probabilities and constant Poisson emissions. We then add the GLM component, which allows us to use inputs, i.e., environmental variables, and the HMM dynamics allow for the relationship between these variables to change over time. However, the influence of the input variables is limited since, unless there is an observation of the output variable, there is no active path between the inputs and the hidden states when we do not observe the outputs. Finally, we implement an input-output HMM architecture a la Bengio and Frasconi [1995], which makes the transition probabilities of our model depend on the inputs.

2 Methods

2.1 Hidden Markov Models

The Hidden Markov Model (HMM) defines a Markov chain on discrete hidden variables $z_{1:T}$. The observed variables $y_{1:T}$ are dependent on the hidden variables through an emission $p(y_t \mid z_t)$. For a set of parameters Θ and $v = 1, \ldots, V$ different events, the HMM joint distribution is

$$p(\{z_{1:T_v}^{(v)}, y_{1:T_v}^{(v)}\}_{v=1}^{V} \mid \Theta) = \prod_{v=1}^{V} \left[p(z_1^{(v)} \mid \Theta) \prod_{t=2}^{T_v} p(z_t^{(v)} \mid z_{t-1}^{(v)}, \Theta) \prod_{t=1}^{T_v} p(y_t^{(v)} \mid z_t^{(v)}, \Theta) \right] \tag{1}$$

We can find the parameters Θ that maximize the marginal log-likelihood of the data by using the Expectation-Maximization algorithm.

2.1.1 Poisson HMM

The Poisson HMM used in this project can characterized by the following distributions:

$$\begin{aligned} z_t \mid z_{t-1}, \{\boldsymbol{\theta}_k\}_{k=1}^{K} &\sim Discrete(\boldsymbol{\theta}_{z_{t-1}}) \\ y_t \mid \{\mu_k\}_{k=1}^{K} &\sim Poisson(\mu_{z_t}) \end{aligned}$$

Where $\boldsymbol{\theta}_k$ and μ_k are the transition probabilities and the scalar Poisson rate, respectively. In this case the data is $\mathcal{D} = \{y_{1:T_v}^{(v)}\}_{v=1}^{V}$ and the log-likelihood comes directly form equation (1).

2.1.2 Poisson GLM HMM

For the GLM we can just replace the Poisson distribution with a Poisson GLM:

$$z_t \mid z_{t-1}, \{\boldsymbol{\theta}_k\}_{k=1}^{K} \sim Discrete(\boldsymbol{\theta}_{z_{t-1}}) \tag{2}$$

$$y_t \mid z_t, \mathbf{x}_t, \{\boldsymbol{\mu}_k\}_{k=1}^{K} \sim Poisson(\boldsymbol{\mu}_{z_t}^T \mathbf{x}_t) \tag{3}$$

In this case $\boldsymbol{\mu}_k \in \mathbb{R}^r$ are regression coefficients of state $k = 1, \ldots, K$. In this case, the data is $\mathcal{D} = \{y_{1:T_v}^{(v)}, x_{1:T_v}\}_{v=1}^{V}$ and we can maximize the marginal likelihood conditional on $\{x_{1:T_v}\}_{v=1}^{V}$. This requires updating the output likelihoods to $p(y_t^{(v)} \mid z_t^{(v)}, \mathbf{x}_t^{(v)} \mid \Theta)$ in equation 1.

2.1.3 Poisson Input-Output HMM

The Poisson Input-Output HMM is a recurrent architecture inspired by the work Bengio and Frasconi [1995]. The authors consider a discrete state dynamical system based on the following state space description:

$$\begin{aligned} z_t &= f(z_{t-1}, \mathbf{x}_t) \\ y_t &= g(z_t, \mathbf{x}_t) \end{aligned}$$

where, same as before, $x_t \in \mathbb{R}^r$ is our input vector at time t, y_t the output values, and $z_t \in \{1, 2, \ldots, k\}$ is the discrete state. This system can be modeled by a recurrent architecture composed by a set of state networks $\mathcal{N}_k$, $k = 1, \ldots, K$ and a set of output networks $\mathcal{O}_k$, $k = 1, \ldots, K$. Each one of the state and output networks is uniquely associated with one of the states, and all networks share the same input $\mathbf{x}_t$. Each state network $\mathcal{N}_k$ has the task of predicting the next state distribution, based on the current input and given that $z_{t-1} = k$. Similarly, each output network $\mathcal{O}_j$ predicts the output of the system, given the current state and input. For this implementation, the output of the system is count data and in line with the rest of the project, we use a Poisson output density.

All the subnetworks are assumed to be static and they are defined by means of smooth mappings $\boldsymbol{\psi}_{kt} = N_k(\mathbf{x}_t; \boldsymbol{\theta}_k)$ and $\eta_{k,t} = O_k(\mathbf{x}_t, \boldsymbol{\vartheta}_k)$, where $\boldsymbol{\theta}_k$ and $\boldsymbol{\vartheta}_k$ are vectors of adjustable parameters. We use the softmax function in the last layer of the state network. In this way, the outputs $\boldsymbol{\psi}_{kt} = \{\psi_{ik,t}\}_{i=1}^{K}$ of the state subnetwork $\mathcal{N}_k$ at time t sum to 1. To ensure that the rates of the Poisson distribution $\eta_{k,t} = \mathbb{E}[y_t \mid z_t = k, \mathbf{x}_t]$ associated with each state k have a positive value, we use the exponential function in the last layer of the output subnetwork $\mathcal{O}_k$. Under this architecture, the distribution of our hidden states and our observation is

$$z_t \mid z_{t-1}, \mathbf{x}_t, \{\boldsymbol{\theta}_k\}_{k=1}^K \sim Discrete(N_{z_{t-1}}(\mathbf{x}_t; \boldsymbol{\theta}_{z_{t-1}})) \tag{4}$$

$$y_t \mid z_t, \mathbf{x}_t, \{\boldsymbol{\vartheta}_k\}_{k=1}^K \sim Poisson(O_{z_t}(\mathbf{x}_t; \boldsymbol{\vartheta}_{z_t})) \tag{5}$$

2.2 Learning the IOHMM

The original model name comes from the fact that it can be used to learn to map input sequences to output sequences, unlike HMMs, which learn the output sequence distribution. For this reason, the authors use EM in a supervised fashion. However, in this case we are interested in the output sequence distribution, so we adapted their model and learning algorithm for unsupervised learning. The EM learning algorithm maximizes the HMM log-likelihood for the complete data $\mathcal{D}_c = \{y_{1:T_v}^{(v)}, z_{1:T_v}^{(v)} \mathbf{x}_{1:T_v}^{(v)}\}_{v=1}^V$:

$$\mathcal{L}_c(\Theta; \mathcal{D}_c) = \log p(\{z_{1:T_v}^{(v)}, y_{1:T_v}^{(v)}\}_{v=1}^V \mid \{\mathbf{x}_{1:T_v}^{(v)}\}_{v=1}^V, \Theta) \tag{6}$$

This requires updating the output likelihoods and transition probabilities to $p(y_t^{(v)} \mid z_t^{(v)}, \mathbf{x}_t^{(v)} \Theta)$ and $p(z_t^{(v)} \mid z_{t-1}^{(v)}, \mathbf{x}_t^{(v)}, \Theta)$ in equation 1. Note that former is the Poisson density with parameter $\eta_{z_t^{(v)},t} = O_{z_t^{(v)}}(\mathbf{x}_t^{(v)}; \boldsymbol{\vartheta}_{z_t^{(v)}})$ and the latter is the state transition probability to $z_t^{(v)}$ given by $\boldsymbol{\psi}_{z_{t-1}^{(v)},t} = N_{z_{t-1}^{(v)}}(\mathbf{x}_t^{(v)}; \boldsymbol{\theta}_{z_{t-1}^{(v)}})$.

In terms of the learning algorithm, let remember that EM replaces the MLE optimization by introducing the auxiliary function $Q(\Theta; \hat{\Theta})$ and iterating the following two steps:

Estimation:	Compute $Q(\Theta; \hat{\Theta}) = \mathbb{E}\left[\mathcal{L}_c(\Theta; \mathcal{D}_c) \mid \mathcal{D}, \hat{\Theta}\right]$
Maximization	Update the parameters as $\hat{\Theta} \leftarrow \arg\max_\Theta Q(\Theta; \hat{\Theta})$

The expectation of (6) can be expressed as

$$Q(\Theta; \hat{\Theta}) = \sum_{v=1}^{V} \sum_{t=1}^{T_v} \sum_{i=1}^{K} \hat{\zeta}_{it} \log p(y_t^{(v)} \mid \eta_{it}) + \sum_{j=1}^{K} \hat{h}_{ij,t} \log \psi_{ij,t}^{(v)} \tag{7}$$

In the original paper they do not use the forward-backward algorithm (smoothing) to compute the posterior marginal probability of state k at time t in the estimation step. Instead they uses the estimated transition probabilities $\psi_{ik,t}$ to propagate an internal state distribution $\boldsymbol{\zeta}_t$. They do use the forward-backward algorithm to estimate the marginal transition probabilities $h_{ik,t}$, however, the leave out the output density at time t. For this implementation we do use the forward-backward probabilities to estimate the marginal probabilities for each event v. For simplicity in notation we omitted the dependence on Θ and the event v. The forward-backward updates are:

$$p(z_t \mid y_{1:T}, x_{1:T}) \propto \alpha(z_t) = p(y_t \mid z_t, \mathbf{x}_t) \sum_{z_{t-1}} p(z_t \mid z_{t-1}, \mathbf{x}_t) \alpha(z_t - 1) \tag{8}$$

$$p(y_{t:T} \mid z_t, x_{t:T}) = \beta(z_{t-1}) = \sum_{z_t} p(y_t \mid z_t, \mathbf{x}_t) p(z_t \mid z_{t-1}, \mathbf{x}_t) \beta(z_t) \tag{9}$$

$$p(z_t, y_{1:T} \mid \mathbf{x}_{1:T}) = \zeta_{z_t,t} \propto \alpha(z_t)\beta(z_t) \tag{10}$$

$$p(z_t, z_{t-1}, y_{1:T} \mid \mathbf{x}_{1:T}) = h_{z_t z_{t-1},t} \propto \alpha(z_{t-1}) p(y_t \mid z_t, \mathbf{x}_t) p(z_t \mid z_{t-1}, \mathbf{x}_t) \beta(z_t) \tag{11}$$

$$\tag{12}$$

We use these equations to update $\hat{\zeta}_{it}$ and $\hat{h}_{ij,t}$ and then proceed to maximize equation (6) by backpropagation.

2.3 Forecasting

The ultimate goal of this project is to forecast the spread of dengue infections. We can produce forecasts from HMM models by Monte Carlo (MC) simulations that draw from the distribution

$$p(y_{T+m} \mid y_{1:T}, x_{1:T+m})$$

Here for simplicity of notation we omit the parameters Θ. For the Poisson HMM this distribution is equal to

$$p(y_{T+m} \mid y_{1:T}) = \sum_{z_{T+m}} p(y_{T+m} \mid z_{T+m}) p(z_{T+m} \mid y_{1:T}) \tag{13}$$

The probability $p(z_{T+m} \mid y_{1:T})$ can be obtained by obtaining the filtered value $p(z_T \mid y_{1:T})$ and propagating it using the static transition probabilities estimated with the model. Here we are implicitly propagating the state forward to $T + m$ and marginalizing out $y_{T+1:T+m-1}$ which we do not observe. This marginalization is easy because of the conditional independence structure of the model. We can then generate samples sequentially by first sampling the state z_{T+M} and then the emission y_{T+m}. The sampling process is equivalent for the GLM HMM, except that we have to use $\mathbf{x}_{1:T}$ for the filtering and $\mathbf{x}_{T+m}$ for the emission probability

$$p(y_{T+m} \mid y_{1:T}, \mathbf{x}_{1:T+m}) = \sum_{z_{T+m}} p(y_{T+m} \mid z_{T+m}, \mathbf{x}_{T+m}) p(z_{T+m} \mid y_{1:T}, \mathbf{x}_{1:T}) \tag{14}$$

Note that because of the model architecture the observations $\mathbf{x}_{T+1:T+m-1}$ are wasted! This is not the case for IOHMM. Under this architecture we have an active path from the observations to the hidden states, and we can use the forward steps to estimate the probabilities $p(z_{T+m} \mid y_{1:T}, \mathbf{x}_{1:T+m})$. The marginalization of the missing emissions $y_{T+1:T+m-1}$ is still straightforward. For our IOHMM, the the forecast distribution is

$$p(y_{T+m} \mid y_{1:T}, \mathbf{x}_{1:T+m}) = \sum_{z_{T+m}} p(y_{T+m} \mid z_{T+m}, \mathbf{x}_{T+m}) p(z_{T+m} \mid y_{1:T}, \mathbf{x}_{1:T+m}) \tag{15}$$

3 Experiments

The dataset contains weekly data for two cities, San Juan and Iquitos, with data for each spanning 5 and 3 years, respectively. For each city, we have a set of 14 features. The features include changes in temperature, precipitation, humidity, and vegetation. We divide the data set into a training and test set in a 70/30 ratio. The latter is a pure future hold-out, meaning the test data are sequential and non-overlapping with any training data. We apply a log transforms to a subset of precipitation variables with scale issues. We also standardize all features before training any model. We deal with missing values in the inputs and outputs by simple linear interpolation. Since multiple input variables measure similar or correlated phenomena, we use PCA to reduce 14 features to 4 principal components. This embedding explains 98 percent of the variance in the covariates.

We estimate each model for $K = 2, 3, 4$ hidden states. We evaluate each model in four dimensions: (i) in-sample Viterbi MAE, (ii) out-of-sample Viterbi MAE, (iii) out-of-sample marginal log-likelihood, and finally (iv) forecast MAE. The first and second tests evaluate the fit, in terms of MAE, after running the Viterbi algorithm for the estimated parameters, for the training and test sets, respectively. The third test evaluates the marginal log-likelihood $\sum_v \sum_t p(y_t \mid \mathbf{x}_t, \hat{\Theta})$ on the test set. Finally, we estimate the model's forecasting error by an eight-week ahead rolling forecast of the test set. The forecast uses the information in the train set and, as the horizon recedes, uses the filter on the new observations.

4 Discussion

We can start the discussion by looking at the results from the model evaluation, which we summarize in Figure 4. There is a lot of information to unpack from this table. The most interesting results are the marginal log-likelihoods of the GLM HMM model. In general, we can observe that the GLM HMM and the IOHMM do worse than the HMM. This observation is not surprising given that we could expect the former two models to be more prone to overfitting to the train set. What is surprising is that the IOHMM, a more complex model, seems to do less overfitting than the more simple GLM HMM. This result could evidence that the additional flexibility from the dynamic transition probabilities extends beyond the training set.

In terms of MAE, there is a lot of mixed information. The more salient result is that every model does considerably worse for the San Juan data. Most models did worst on the forecasts for San Juan than our baseline model, a factor model for the inputs fed into a ARX model. This is reasonable given that, as can be seen from Figure 5, the San Juan data has a lot

more peaks, both in terms of numbers and severity. The baseline model predicts an eight week lagged observation (Figure 6) which could yield a lower MAE given that the biggest errors seem to accumulate when the model misses big peaks.

The forecasts for HMMs show interesting insight into some limitations of these models in forecasting. For the discussion, we focus on the case of three hidden states. If we look at Figure 7 and 8 we can see the forecasted case counts and the forecasted hidden state distribution, respectively. The count forecasts show a switching behavior between two regimes with different emission rates. If we look at the state distributions, we can see that they are relatively constant except for brief periods where states zero and one increase their likelihood and state's two decrease. The constant state distributions result from the static transition probabilities that quickly converge to the observed distributions, independent of the past cases. Only when there are spikes in cases, which leads to state one becoming very likely, do we see that some of this information can get to the m step forecast.

We can observe slightly different behavior for the GLM HMM in figures 9 and 10. The forecasts seem to be more proactive and the state distributions more static. We can attribute the latter can again to the static transition probabilities. Despite this, the input variable modulation makes them very noisy and hard to interpret.

Finally, the IOHMM in Figure 11 seems to be the most flexible of the HMMs models; yet it is still not able to anticipate large increases in the infection counts. If we look at the state transition distribution from 12 we can see that it is more dynamic; however, it still reacts with lags. Despite the active path between the input variables and the hidden states, the past case counts still have the most valuable information for updating the model beliefs.

5 Conclusion

The project's goal was to build a model that could forecast the spread of dengue infection in Central America. However, this goal quickly turns secondary to exploring a different way to use information in HMMs architectures. Despite our hardest efforts to better use the information, we were not able to do significantly better than the baseline model. In addition to trying different architectures that allow us to integrate more information, it would be good to spend time adding structure to the models that can better use the data. For example, we could explore using moving averages of the environmental variables, as was first suggested in the midway report. Also, we could use tools such as Granger causality to explore actual causal relationships between the environmental variables and the spread of infections. Finally, given the differences in the scale of case counts coming from both cities, we could get better results by introducing a dummy variable for each city. This would still allow us to jointly estimate the parameters of the model using the data from both cities but would allow for systematic differences between them.

The code of the project is available at `https://github.com/jilanglois-su/cobs10-dengai`.

References

DrivenData. Competition: Dengai: Predicting disease spread. `https://www.drivendata.org/competitions/44/dengai-predicting-disease-spread/`, November 2022.

N Nova, ER Deyle, MS Shocket, AJ MacDonald, ML Childs, M Rypdal, G Sugihara, and EA Mordecai. Susceptible host availability modulates climate effects on dengue dynamics. *Ecol Lett.*, 2021.

Yoshua Bengio and Paolo Frasconi. An input output hmm architecture. *Advances in neural information processing systems*, 1995.

Appendix A: Additional Figures

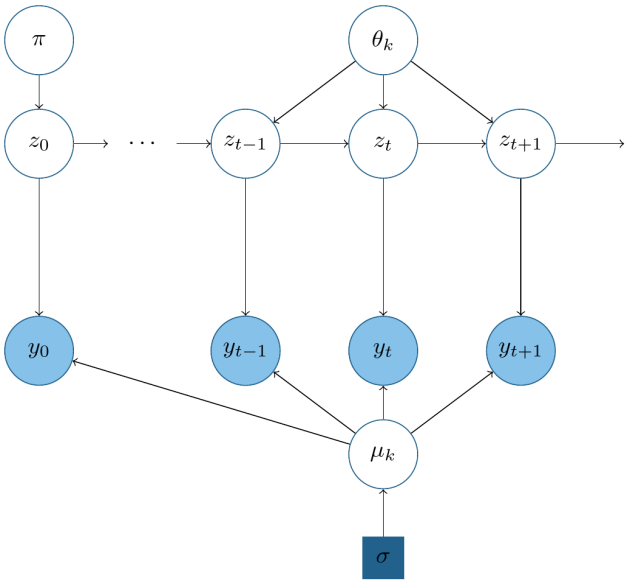

Figure 1: HMM graphical model.

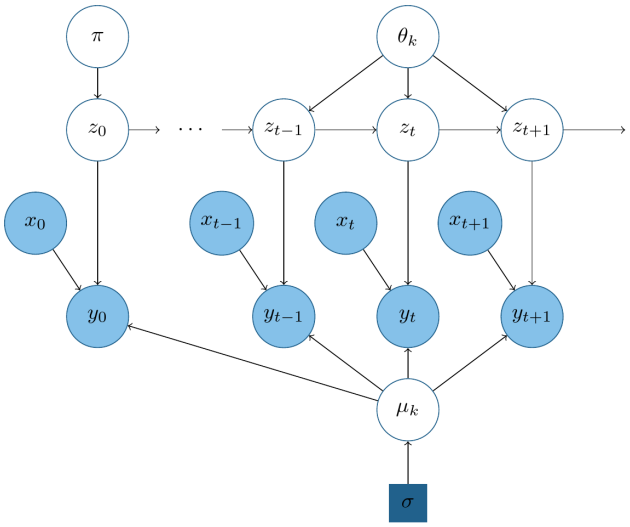

Figure 2: GLM HMM graphical model.

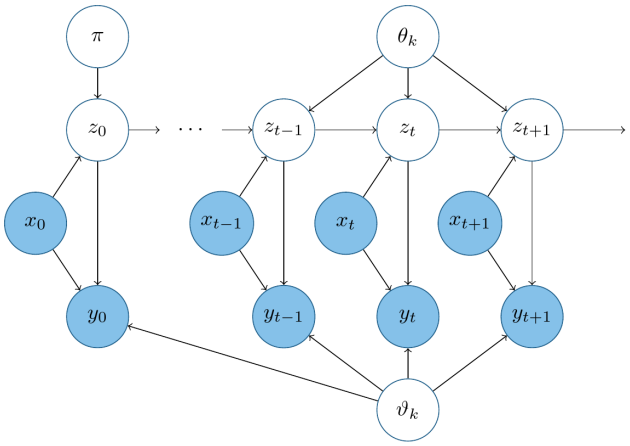

Figure 3: IOHMM graphical model.

Model	Number of States	City	Marginal Log-Likelihood	Forecast MAE	In-Sample MAE	Out-of-Sample MAE
HMM	2	iq	-2893.90	10.05	9.40	9.62
		sj	-2893.90	14.73	22.05	10.02
	3	iq	-2392.93	9.76	4.63	4.37
		sj	-2392.93	15.49	16.44	10.91
	4	iq	-2027.78	10.59	3.57	4.08
		sj	-2027.78	14.80	14.66	9.05
GLM HMM	**2**	iq	-3230.49	7.09	3.85	4.91
		sj	-3230.49	17.54	20.33	15.27
	3	iq	-3017.20	7.81	3.18	4.64
		sj	-3017.20	23.50	16.68	13.16
	4	iq	-3234.52	7.78	2.94	4.90
		sj	-3234.52	32.98	15.17	12.80
IOHMM	2	iq	-3397.03	8.40	4.18	5.07
		sj	-3397.03	28.61	21.05	16.33
	3	iq	-2226.35	7.85	2.77	3.40
		sj	-2226.35	22.07	14.25	10.72
	4	iq	-2117.51	7.73	2.71	3.41
		sj	-2117.51	29.59	10.82	8.81

Figure 4: Model evaluation results.

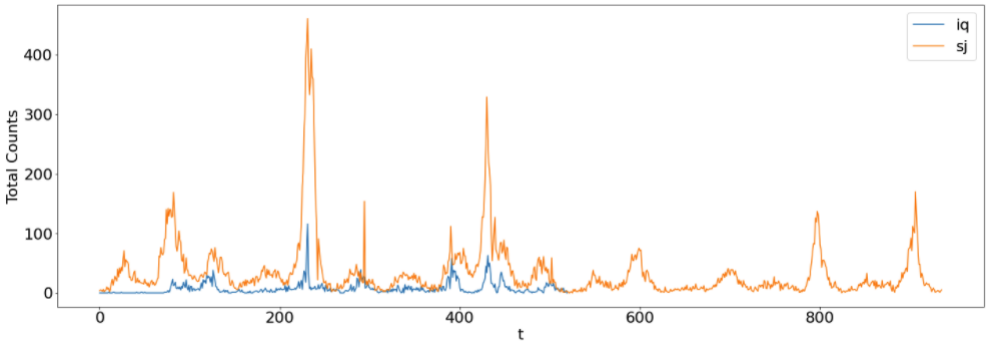

Figure 5: Complete observations (train and test sets) for each city.

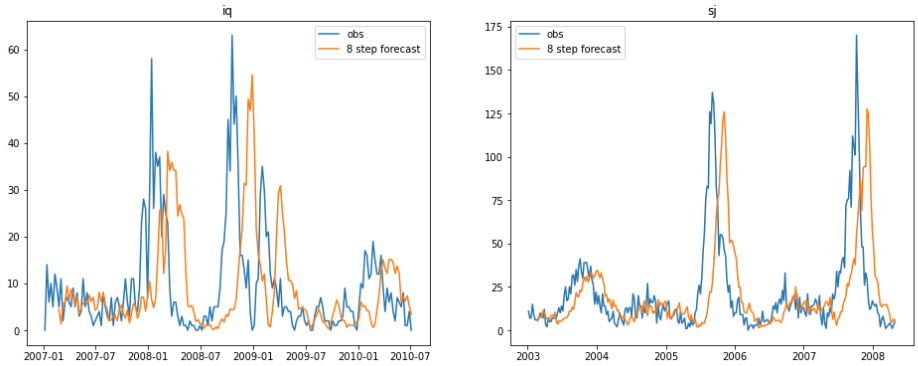

Figure 6: Forecasts baseline model.

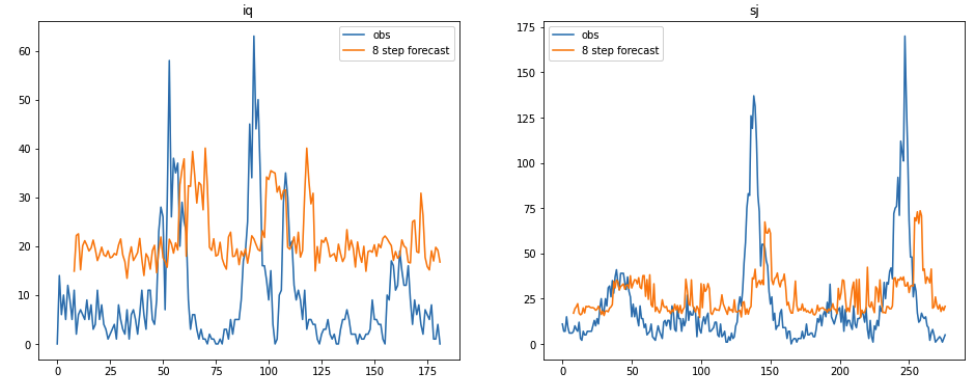

Figure 7: Counts forecasts for the HMM model with 3 hidden states.

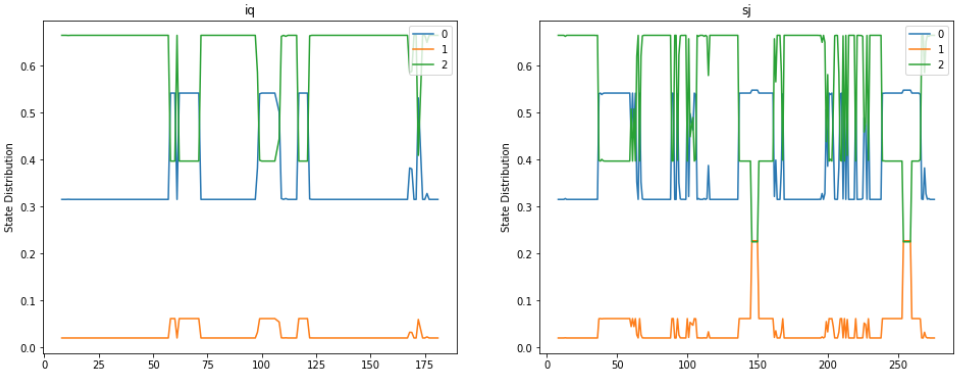

Figure 8: State distribution forecasts for the HMM model with 3 hidden states.

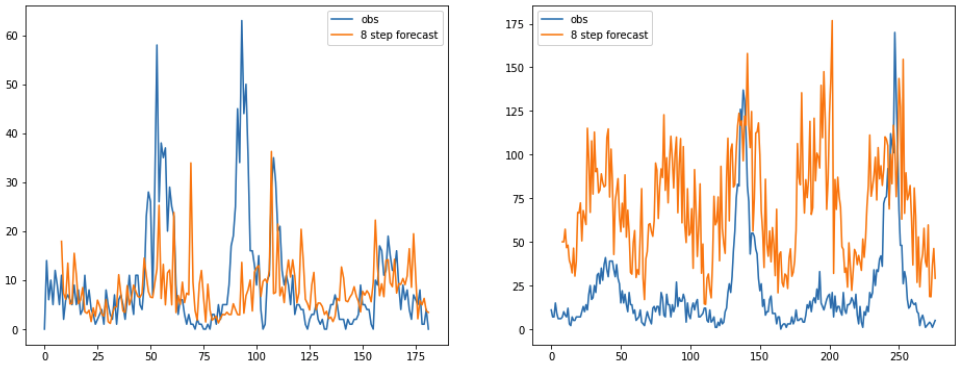

Figure 9: Count forecasts for the GLM HMM model with 3 hidden states.

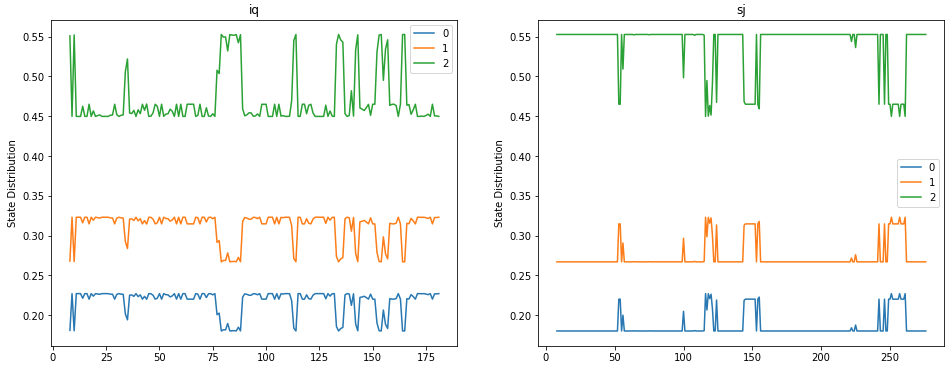

Figure 10: State distribution forecasts for the GLM HMM model with 3 hidden states.

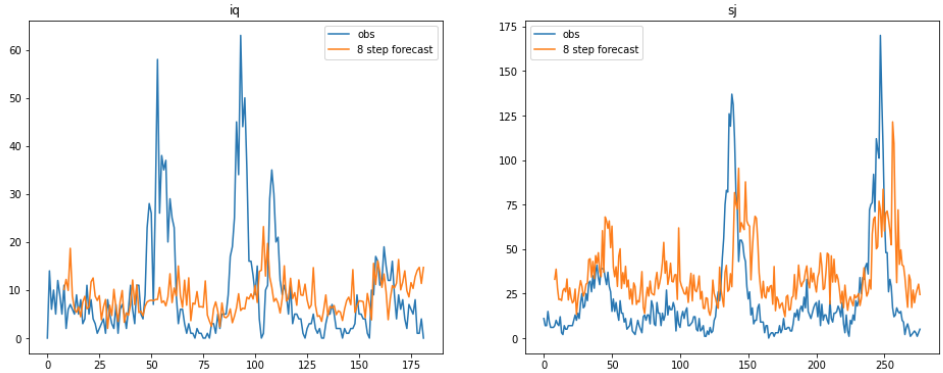

Figure 11: Count forecasts for the IOHMM model with 3 hidden states.

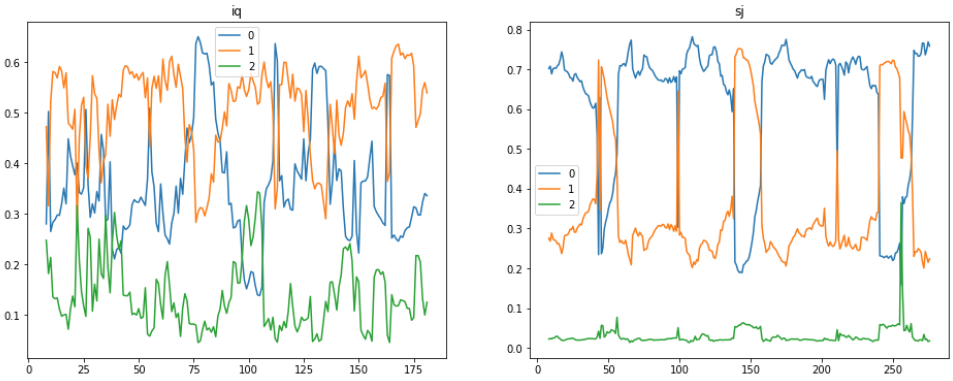

Figure 12: State distribution forecasts for the IOHMM model with 3 hidden states.