LISP: Localized Inference through Selective Pre-training for Critical Diagnosis Forecasting

Karan Samel*[1] and Jimeng Sun†[2]

[1]Department of Computational Science and Engineering, Georgia Tech
[2]Department of Computer Science, University of Illinois at Urbana-Champaign

April 21, 2020

1 Introduction

The quality of care patients receive in hospitals is directly correlated with the medical staff and resources available. To provide such a service, hospital providers have historically used heuristics to determine how to allocate medical staff, such nurses, to each hospital service. A common heuristic is the nursing intensity score, which is a composed of many administrative and care related tasks [1]. These heuristics are typically static and do not accurately predict the necessary workload needed by medical staff. Due to these issues, there is typically administrative overhead and burnout observed with these practitioners [2]. Incorrect patient forecasting requirements also lead to monetary costs of unproductive over staffing, and with under staffing, poor patient care [3]. To improve the mental health of these medical staff, and thus the patients, more dynamic heuristics have been proposed based on feedback from these staff and patient volume [4].

There are many works that explore modeling the patient volume component of these heuristics which is becoming more feasible with the availability of Electronic Health Records (EHRs). These records contain detailed records or diagnosis, procedures, drugs administered for each patient over time. Leveraging this data to predict future patient volume has been explored in both the traditional time series forecasting, and more recently with more complex machine learning models. These machine learning models have proven to extract salient features from complex data sets in order to perform prediction tasks and have been leveraged for time series forecasting.

In the patient forecasting use case, we leverage the patient diagnosis data which is defined through the International Classifications of Diseases (ICD) hierarchy. Our model encodes this hierarchical relation using graph neural networks as part of an end to end architecture for patient forecasting. With this model we provide more accurate patient forecasting for various levels of diagnoses in the hierarchy, ranging from total patient volume to patient volume for a specific diagnosis. While previous work focus on total patient forecasting or a specific diagnosis, we have developed a single modeling framework to *more accurately forecast a broader range of diagnoses using data from multiple providers.*

This is important as it provides provider more granular predictions of diagnosis volume, which can lead to better allocation of health practitioners. This is especially crucial for certain complications and co-morbidity diagnoses which are critical to patients and cost

*ksamel@gatech.edu
†jimeng@illinois.edu

insurers \$31 billion annually [5]. We compare our model to the traditional methods used. Such exploration can better prepare medical staff to handle these diagnoses and minimize these incurring costs.

2 Related Works

To mitigate costs of over and under staffing, various time patient forecasting models have been explored at the patient level. There are traditional time series approaches that have been tested using Autoregressive Integrated Moving Average (ARIMA) with various levels of seasonality (SARIMA), and data processing [3]. SARIMA and Simple Exponential Smoothing (SES) models have been tested to predict daily patient volume within hospitals [6] and emergency departments [7] respectively. Averaging of multivariate time series models over patient volume and other temporal factors have also been tested to efficiently [8].

In addition to traditional time series methods, machine learning techniques have been applied to patient forecasting as well. A hybrid approach of wavelet decomposition and neural networks were successfully applied to hospital volume predictions [9]. Other deep learning models including Convolutional Neural Networks (CNNs), Long Short Term Memory networks (LSTMs), and boosted trees through Extreme Gradient Boosting (XGBoost), were tested, where XGBoost provided the best performance over the other machine learning and time series models [10].

With hierarchical structured data such as diagnosis and drug codes, we also explore time series methods that leverage this structure. Hierarchical time series methods that decompose hospital occupancy into their sub-divisons for more targeted analysis [11]. There are works that encode these codes using Graph Neural Network (GNN) techniques to get a salient join representation for each code with respect to its neighbors [12]. These features have also leveraged in the forecasting domain, where each time component can be represented as a graph structure. These works are prominent in traffic forecasting, where the sensor nodes are arranged in a grid like pattern and a temporal structure captures the relationship across these GNN representations [13, 14, 15, 16].

In these traffic prediction use case, the graph structure contains different sensor values at each time period. In our patient and diagnosis forecasting, we leverage co-occurrence relations seen in the EHR data. This is used in turn to model patient volumes for specific critical diagnoses, which provides more actionable insights to support health worker staffing. The diagnosis modeling varies per healthcare provider, and is augmented by available diagnosis patterns from other providers.

3 Problem Formulation

Our objective is to forecast the number of diagnosis y_{dx_t} at time step t given data from T slices of previous patient records. Each of these time slices are aggregated patient data in windows of 7 days, which gives adequate time to plan for staffing distribution. Given batched patient records X_t which contain the volume for all diagnosis during batch t, we want to predict the diagnosis volume y_{dx_t} for one specific diagnosis at t.

This is done by finding our model $f \in R^{TxD} \rightarrow R$. This is a mapping from T slices with D diagnosis counts to a single diagnosis count y_{dx_t}. Our objective function can be defined as follows:

$$A = \{X_{t-1}, ..., X_{t-T}\}$$

$$\hat{f} = \underset{f}{\operatorname{argmin}} ||y_{dx_t} - f(A)||_1$$

We are interested in the L1 loss over L2 to avoid over correcting to large deviations and noise seen in provider data.

4 Method

To leverage historical diagnosis information, a co-occurrence graph is built over the EHR data. This representation is used by a hybrid temporal GNN model with an Autoregressive component. Our model, LISP, is first pre-trained using data from certain providers (Selective Pre-training) and fine tuned for the provider of interest (Localized Inference).

4.1 Diagnosis Co-occurrences

To represent diagnosis codes in a graphical fashion, a starting point is to use the ICD hierarchy. This taxonomy of diseases is useful for insurance and billing purposes, where sub-classes of diagnosis can be identified quickly. For the purposes of forecasting, it is imperative to find relationships between diagnosis that have a direct correlation with their observation volume. This does not directly occur within the ICD hierarchy, as neighboring diagnosis are structurally similar but may not be predictive of one another.

The co-occurrence graph is built as follows in order to limit the computational complexity during the model training process. We choose a subset of diagnosis dx_s to include in our graph account for α % of the overall cumulative distribution of all diagnosis dx in the EHR data.

$$\alpha = \sum_{t \in dx_s} f_{dx}(t)$$

Our selected dx_s determine the nodes used in our graph. For each diagnosis in dx_s, a similar thresholding is conducted to limit the number of co-occurrence edges. For each time slice in our data, the feature for each node is the count for that diagnosis within the time slice. Each node feature are further normalized over the maximum occurrence of those features over all time slices.

4.2 Temporal GCN Model

Given historical diagnosis information arranged in a structural manner. It is important to define a model architecture that can leverage this information. The family of GNN models have been proven to efficiently encode the graph node representations for end to end training. With the normalized feature counts, we use a Graph Convolutional Network (GCN) [17] to encode each time slice into a higher dimensional representation vector x_h. This leverages the node feature x_v and aggregates it with its neighborhood $N(v)$ with a neighborhood normalization. For end to end training, this representation is multiplied with weight parameters W and following by a non-linearity.

$$h_v = \sigma(W \sum_{u \in \{N(v), v\}} \frac{x_u}{\sqrt{|N(u)||N(v)|}})$$

This representation can be repeated using h_v^l to produce a higher level representation h_v^{l+1} which can be interpreted as taking a larger hop neighborhood, as the convolution is occurring on neighborhood representations that encode their neighborhoods in the previous step.

Our model comprises of a GCN layer, followed by a 1-D CNN layer to compute temporal features. This temporal representation is followed by another GCN layer and then is followed by fully connected layers to predict the diagnosis volume.

4.3 Hybrid Approach

When training the temporal GCN on data from a single provider using a 7 day time window, the number of data points may only be in the the 100's. Compared to traffic forecasting datasets which have 10,000's of unique samples [15], using the same complex architecture will overfit on our provider data, producing high variance estimates. To mitigate this effect, an Autoregressive (AR) component is integrated into the model. This is done by adding the predictions of the temporal GCN model with the AR output [18], which helps control this variance. The AR component directly uses the historical values of the diagnosis of interest $\{y_{dx_{t-1}}, ..., y_{dx_{t-T}}\}$ as it's input.

4.4 Selective Pretraining

With the hybrid model in place will still have a difficult time generalizing past the training set due the limited number of data points per provider. The GCN can better converge to the underlying co-occurrence distribution by training on other providers as well. The key limitation that prevents pre-training on any provider is that different providers will have different distributions within their weekly diagnoses counts.

To address this, a different subset of providers must be taken for each diagnosis model. The distribution diagnosis of providers must be compared with repect to our prediction diagnosis. This is done by constructing a set of top K most frequent diagnosis within a provider's p EHR dataset, denoted as dx_K^p. For our diagnosis of interest dx we also want to have its co-occurring values dx_c in within the provider EHR such that $dx_C^p = dx \cup dx_c$. Given these sets, we select our dx specific providers P_{dx} from all providers P_{all} as follows:

$$P_{dx} = \{p \in P_{all}; \frac{|dx_K^p \cap dx_C^p|}{|dx_C^p|} > \beta\}$$

Where β is our overlap threshold. With $\beta \approx 1$ we identify that all co-occurring diagnosis must be present, but are given fewer providers. On the opposite end, $\beta \approx 0$ means that few co-occurring diagnosis signals are needed, thus more provider data sets are returned. In practice $\beta \in [0.2, 0.6]$ works well, with lower β better supporting less common diagnoses.

Once the model has been pre-trained on the selected providers, it is trained on the data from the provider of interest for fine tuning.

5 Experiments

5.1 Data

We use a data set provided by Truven Health Analytics, which contains health insurance data for inpatient and outpatient cases form multiple providers from 2011 to 2015. Each patient admission case contains the diagnosis for that visit as well as other patient specific features and were timestamped per day. The diagnosis codes were converted to ICD-10 CM.

Based on the volume of diagnoses, other ICD codes within the hierarchy, including complication and co-morbidity codes were selected for forecasting. We monitor the forecasts for metabolic disorders and a complication code for cerebral infractions.

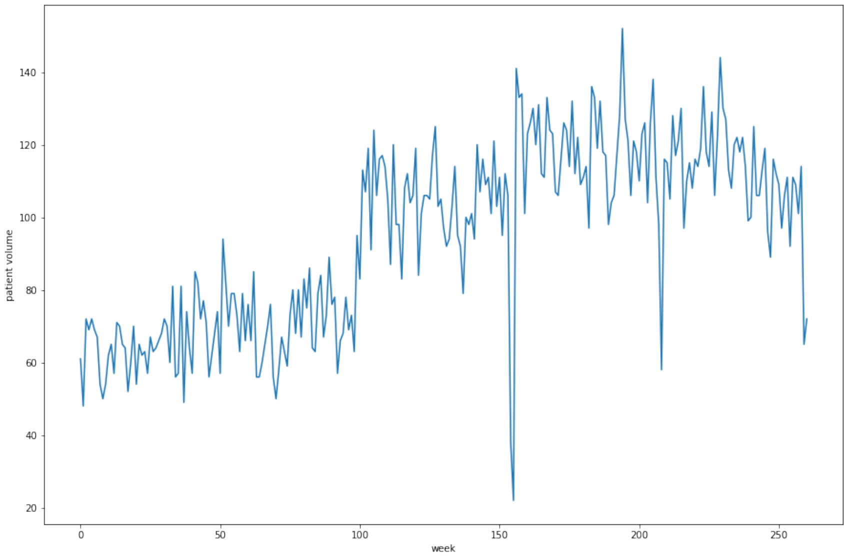

Figure 1: Here is the weekly patient volume at a certain provider. There are no clear trends as and the mean of the distribution changes over time. There are also large deviations or outliers based on the state of the provider and due to noise in the data set.

We focused our analysis on a single provider where the with 70%, 20%, and 10% of the data for any provider were utilized for training, validation, and a holdout test set for final evaluation.

5.2 Data Transformations

When observing the diagnosis counts in providers, it is difficult to observe any seasonal or cyclical trends using week window granularity (Figure 1). Due to the many factors that affect the diagnosis volume for a provider, it is common in time series task to perform a data normalization task to transform the target values to follow a normal distribution. With this, a model is tasked with predicting the variance at a certain time step and the transformation can be reversed to follow the original distribution.

For this transformation we used a Box-Cox transformation, a power transform which handles positive values. Since it is possible to have a 0 count for a diagnosis in a time slice, we modified the transformation with a constant offset $\epsilon = 0.001$. Given the original diagnosis count y_dx the transformed value becomes y^t_{dx} and vice versa using the inverse transform.

$$y^t_{dx} = \begin{cases} \log(y_{dx} + \epsilon) & \text{if } \lambda = 0 \\ \frac{y^\lambda_{dx} + \epsilon - 1}{\lambda} & \text{otherwise} \end{cases} \tag{1}$$

λ is computed through the MLE over a Gaussian distribution. To invert the transform, a the inverse formulation follows.

$$y_{dx} = \begin{cases} e^{y^t_{dx}} - \epsilon & \text{if } \lambda = 0 \\ (\lambda y^t_{dx} + 1)^{\frac{1}{\lambda}} - \epsilon & \text{otherwise} \end{cases} \tag{2}$$

In addition to a power transform, we also want to have a constant mean, which is also not present in most forecast data. To accomplish this the target predictions were differenced, where the difference between two consecutive y^t_{dx} were taken to generate the target distribution that is trained on y^{td}_{dx}. When an inference from a model is made, the predction must be de-differenced based on diagnosis volume value form the last time step and then the inverse Box-Cox is computed to provide a prediction $\hat{y_{dx}}$resembling the original y_{dx}.

5.3 Baselines

Based on the previous literature on patient forecasting [10], we select the following baselines to compare our results to: we compared our approach to the Seasonal ARIMA method, which proves robust in high variance forecasting tasks. For SARIMA, a Box-Cox Transformation was applied on the training data, and was differenced by 1. This was then fed into a grid-search for parameter tuning.

We are interested in finding out how our model compares to these baseline methods at patient and diagnosis forecasting at different levels.

5.4 Metrics

We measure the error between the true volume y_k and our forecasted value f_k as $e_k = y_k - f_k$ for the kth item in the test set. We measure this through the root mean square error (RMSE) and Theil's U which are defined as follows:

$$RMSE = \sqrt{\frac{1}{K}\sum_{k=1}^{K} e_k^2}$$

$$Thiel's\ U = \frac{\sqrt{\frac{1}{K}\sum_{k=1}^{K} e_k^2}}{\sqrt{\frac{1}{K}\sum_{k=1}^{K} f_k^2}\sqrt{\frac{1}{K}\sum_{k=1}^{K} y_k^2}}$$

RMSE measure the absolute squared error, while Thiel's U normalizes this error with respect to the size of the forecasts. RMSE is the traditional approach when comparing methods within a single target data set. Thiel's U performance is invariant of the data distribution, which is more useful when evaluating performance over multiple data sets.

5.5 Results

For our evaluation we test our method on overall patient volume and two other critical diagnosis. For each case since there are only a few test data points for each selected provider, we take the average metrics across the top 10 largest providers used for each task. We compare results across all diagnosis codes and then codes that are more specific, with less data available.

Method	RMSE	Thiel's U
SARIMA	1.526947	4.182187
LISP	1.546738	2.383069

Table 1: Results for predicting overall patient volume.

Method	RMSE	Thiel's U
SARIMA	1.774082	4.586198
LISP	1.973802	0.691751

Table 2: Results for code J96.90: Respiratory failure, unspecified.

SARIMA slightly outperforms LISP with respect to RMSE and it is able to capture the average volume well. However as volume data between different sources is heteroskedastic, LISP better captures these features in the data, reducing the U statistic.

Method	RMSE	Thiel's U
SARIMA	1.764163	1.022248
LISP	1.956040	0.654381

Table 3: Results for code J18.9: Pneumonia, unspecified organism.

6 Conclusion

In general, we see that the traditional SARIMA is quite robust at predicting patient volumes at different levels of the ICD hierarchy. With the lack of temporal data slices, LISP attempts to aggregate code frequencies across different providers and normalize them. This combined with an AR output helps control the variance of the deep learning model.

In the future, we would like to try out other deep learning baselines such as LSTM, Xgboost, and the original STGCN.

References

[1] Isabelle Briatte, Caroline Allix-Béguec, Gérard Garnier, and Mercédès Michel. Revision of hospital work organization using nurse and healthcare assistant workload indicators as decision aid tools. *BMC health services research*, 19(1):554, 2019.

[2] WFJM van den Oetelaar, HF Van Stel, W Van Rhenen, RK Stellato, and W Grolman. Balancing nurses' workload in hospital wards: study protocol of developing a method to manage workload. *BMJ open*, 6(11):e012148, 2016.

[3] Ireneous N Soyiri and Daniel D Reidpath. An overview of health forecasting. *Environmental health and preventive medicine*, 18(1):1, 2013.

[4] Shu Yin Hoi, Norafida Ismail, Li Chern Ong, and Jasmine Kang. Determining nurse staffing needs: the workload intensity measurement system. *Journal of nursing management*, 18(1):44–53, 2010.

[5] Jessica Kent. Machine learning, ehr data predict high-risk surgical patients, Dec 2018.

[6] Li Luo, Le Luo, Xinli Zhang, and Xiaoli He. Hospital daily outpatient visits forecasting using a combinatorial model based on arima and ses models. *BMC health services research*, 17(1):469, 2017.

[7] Hye Jin Kam, Jin Ok Sung, and Rae Woong Park. Prediction of daily patient numbers for a regional emergency medical center using time series analysis. *Healthcare informatics research*, 16(3):158–165, 2010.

[8] Stephen DeLurgio, Brian Denton, Rosa L Cabanela, Sandra Bruggeman, Arthur R Williams, Sarah Ward, Ned Groves, and John Osborn. Forecasting weekly outpatient demands at clinics within a large medical center. *Production and Inventory Management Journal*, 45(2):35–46, 2009.

[9] Lean Yu, Geye Hang, Ling Tang, Yang Zhao, and KK Lai. Forecasting patient visits to hospitals using a wd&ann-based decomposition and ensemble model. *Eurasia Journal of Mathematics, Science and Technology Education*, 13(12):7615–7627, 2017.

[10] Brian Klute, Andrew Homb, Wei Chen, and Aaron Stelpflug. Predicting outpatient appointment demand using machine learning and traditional methods. *Journal of medical systems*, 43(9):288, 2019.

[11] Christoph Weiss. *Essays in Hierarchical Time Series Forecasting and Forecast Combination.* PhD thesis, University of Cambridge, 2018.

[12] Alessandro Sperduti and Antonina Starita. Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3):714–735, 1997.

[13] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.

[14] Jiani Zhang, Xingjian Shi, Junyuan Xie, Hao Ma, Irwin King, and Dit-Yan Yeung. Gaan: Gated attention networks for learning on large and spatiotemporal graphs. *arXiv preprint arXiv:1803.07294*, 2018.

[15] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3634–3640. AAAI Press, 2018.

[16] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. 2019.

[17] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[18] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 95–104. ACM, 2018.