Wikipedia Traffic Forecasting with Graph Embeddings

Anthony Miyaguchi, Shaon Chakrabarti, and Nicolai Garcia

CS229 - Fall 2019

Summary

We build a model to forecast the number of visits to a random, connected subset of Wikipedia articles over a 7-day window.

- We mine unsupervised features from article hyperlinks using spectral graph methods.
- Through ablation analysis on n=10 trials, we find no significant improvement in our model performance when incorporating these features.

Introduction

Web traffic patterns evolve in response to information demand. Models of these patterns can be used to analyze trends across pages or to forecast future demand. Can the hyperlinked structure of Wikipedia be incorporated to improve performance of machine learning models of article traffic?

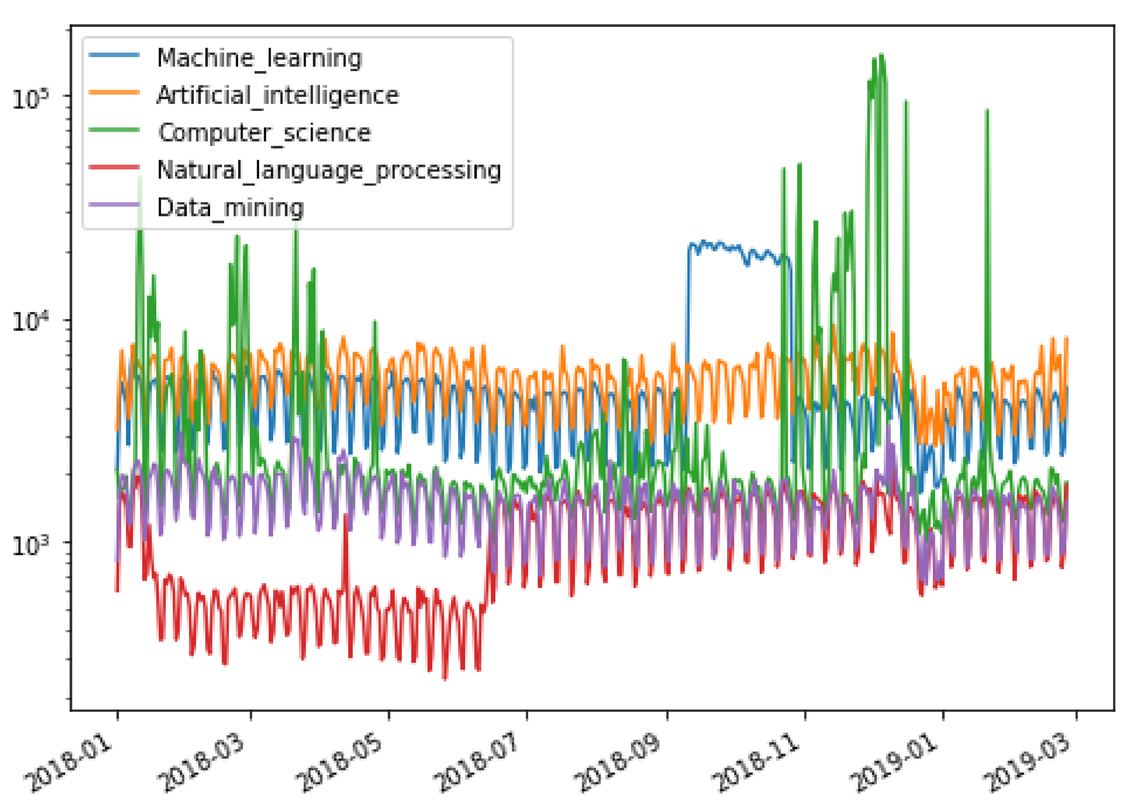

Figure 1:Traffic of articles related to `Machine_learning`.

Data

The data is processed using [4] on the Wikipedia SQL dump from August 2019 and 60 weeks of page view data from 2018-01 to 2019-03. There are 5.9×10^6 pages and 4.9×10^8 directed hyperlinks. We randomly sample a fraction of articles, compute the induced graph, and use the largest connected component for experimentation.

Experiments and Features

In experiment 1, we observe random, connected graphs with size $|V| = 3.7 \times 10^4$ and $|E| = 1.7 \times 10^5$ and compute PageRank and a spectral embedding.[2]. These are concatenated with the historical page views.

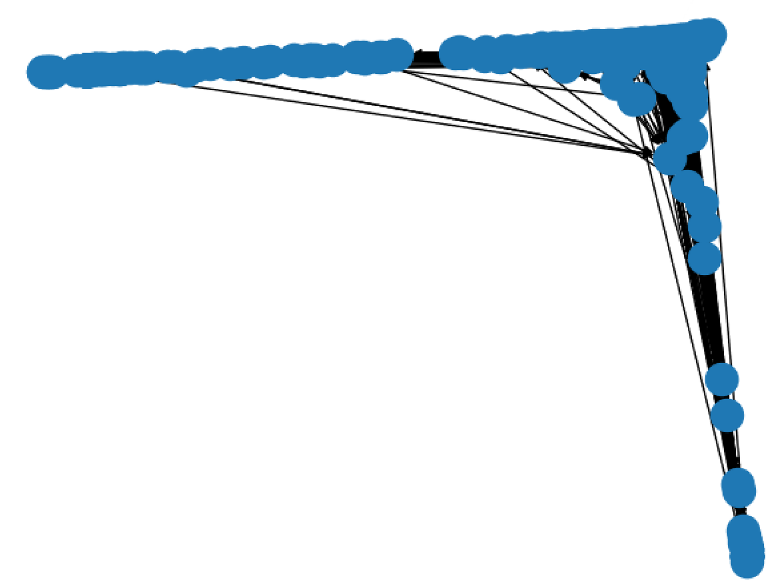

Figure 2:Spectral embedding of a sampled sub-graph.

In experiment 2, we recursively cluster a sub-graph of size $|V| = 2.9 \times 10^6$ and $|E| = 5.2 \times 10^7$ to generate sign and vector features.

```
-RECORD 0------------------------------
 id           | 16437
 partition_id | ooxxoooo
 title        | Reinforcement_learning
 degree       | 288
 inDegree     | 135
 outDegree    | 153
 sign_0       | false
 fiedler_0    | -1.8320626091157155E-6
 2018-01-01   | null
 2018-01-02   | 101
 2018-01-03   | null
```

Figure 3:Sample record from experiment 2 with metadata, degree, sign, vector, and page view columns.

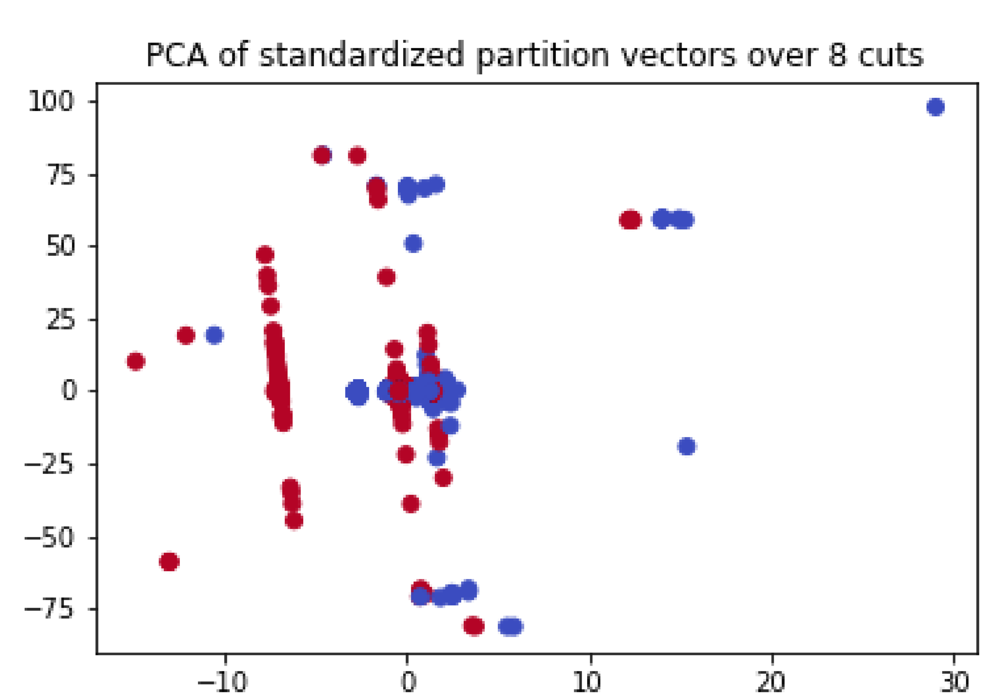

Figure 4:Recursively-mined partition vectors [1] approximates an optimal ratio-cut and generates unsupervised labels.

Models

We report the performance of the following regression models for page view traffic.

Persistence	$\hat{y}_{t_0}^{(i)} = x_{(t_0-T)}^{(i)}$
Mean	$\hat{y}_{t_0}^{(i)} = \frac{1}{T}\sum_t^T x_{(t_0-T+t)}^{(i)}$
Linear Regression	$\hat{y}_{t_0}^{(i)} = \sum_{t=1}^T \theta^T x_{(t_0-T+t)}^{(i)} - \epsilon_{t_0}$
Ridge Regression	Linear model with L_2 regularization
Neural Network	Fully connected, feedforward, L_2 regularization
TRMF[3]	Latent space model using matrix methods

Metrics

Mean Absolute Precision Error (MAPE):

$$MAPE = \left[\frac{1}{nT}\sum_{i=1}^{n}\sum_{t=1}^{T}\left|\frac{Y_t^{(i)} - Z_t^{(i)}}{Y_t^{(i)}}\right|\right] * 100\% \quad (1)$$

Normalized Root Mean Square Error (NRMSE):

$$NRMSE = \frac{1}{\sum_i^n \sum_t^T Y_t^{(i)}}\sqrt{\frac{1}{nT}\sum_{i=1}^{n}\sum_{t=1}^{T}(Y_t^{(i)} - Z_t^{(i)})^2} \quad (2)$$

Results

All results were gathered from 10 trials.

	Exp 1		Exp 2	
model	NRMSE	MAPE	NRMSE	MAPE
persistence	7.2 ± 1	7.0 ± 0	7.5 ± 0	8.4 ± 0
mean	5.7 ± 0.6	7.2 ± 0	6.1 ± 0	8.9 ± 0
linear reg.	6.1 ± 0.6	7.6 ± 0.3	5.9 ± 0	7.7 ± 0
ridge reg.	6.0 ± 0.6	7.5 ± 0.3	6.2 ± 0	8.5 ± 0
neural network	5.9 ± 0.7	6.9 ± 0.3	5.9 ± 0	9.1 ± 0.3
trmf	-	-	6.5 ± 0	8.6 ± 0.1

Ablation Analysis

Experiment 1	NRMSE	MAPE
neural network	5.9 ± 1.0	6.9 ± 0.9
without history	0.2 ± 0.6	-0.4 ± 0.4
without pagerank	0.1 ± 0.4	0.0 ± 0.4
without embedding	0.0 ± 0.4	0.0 ± 0.4

Experiment 2	NRMSE	MAPE
neural network	5.9 ± 0.1	9.1 ± 1.0
without history	0.1 ± 0.0	0.9 ± 0.4
without degree	0.0 ± 0.1	0.0 ± 0.6
without sign	0.1 ± 0.0	0.7 ± 0.4
without vector	0.1 ± 0.1	0.6 ± 0.4

Discussion

We were not able to build a model that consistently outperformed simple persistence and linear models. This may be due to limited resolution of the data, as shown by lackluster performance of TRMF. We determined that the embedding features do not improve performance at daily granularity. In the second experiment, we note some reliance on graph features by observing the greater increase in MAPE by dropping historical page views than the sign or vector features.

Future

We would proceed next by building a convolutional neural network to exploit symmetries in the time and graph signals, as well as forecasting over a longer window (months) or higher frequency (hours). Additionally, we would attempt to scale the first few cuts of our partitioning implementation increase our training set for non-linear models.

References

[1] L. Hagen and A. B. Kahng, "New spectral methods for ratio cut partitioning and clustering," *IEEE transactions on computer-aided design of integrated circuits and systems*, vol. 11, no. 9, pp. 1074–1085, 1992.

[2] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in neural information processing systems*, 2002, pp. 585–591.

[3] H.-F. Yu, N. Rao, and I. S. Dhillon, "Temporal regularized matrix factorization for high-dimensional time series prediction," in *Advances in neural information processing systems*, 2016, pp. 847–855.

[4] N. Aspert, V. Miz, B. Ricaud, and P. Vandergheynst, "A graph-structured dataset for wikipedia research," in *Companion Proceedings of The 2019 World Wide Web Conference*, ACM, 2019, pp. 1188–1193.