A Flexible Forecasting Framework for Hierarchical Time Series with Seasonal Patterns: A Case Study of Web Traffic

Zitao Liu*
Pinterest
651 Brannan St
San Francisco, California 94107
zitaoliu@pinterest.com

Yan Yan†
Intelligent Advertising Lab, JD.COM
675 E Middlefield Rd
Mountain View, California 94043
zitaoliu@pinterest.com

Milos Hauskrecht
Department of Computer Science
University of Pittsburgh
Pittsburgh, Pennsylvania 15260
milos@cs.pitt.edu

ABSTRACT

In this work, we focus on models and analysis of multivariate time series data that are organized in hierarchies. Such time series are referred to as hierarchical time series (HTS) and they are very common in business, management, energy consumption, social networks, or web traffic modeling and analysis. We propose a new flexible hierarchical forecasting framework, that takes advantage of the hierarchical relational structure to predict individual time series. Our new forecasting framework is able to (1) handle HTS modeling and forecasting problems; (2) make accurate forecasting for HTS with seasonal patterns; (3) incorporate various individual forecasting models and combine heuristics based on the HTS datasets' own characterization. The proposed framework is evaluated on a real-world web traffic data set. The results demonstrate that our approach is superior when applied to hierarchical web traffic prediction problems, and it outperforms alternative time series prediction models in terms of accuracy.

KEYWORDS

Multivariate time series; Forecasting

1 INTRODUCTION

Many organizations in business, economics or information technology operate in a multi-item, multi-level environment. Time series data representing the behaviors of these organizations can be often organized in a hierarchical (tree) structure where different time series interact and influence each other. These related multivariate time series are referred to in the literature as *hierarchical time series* (HTS) [6].

Individual time series within the same hierarchy not only interact and correlate with each other, but they often satisfy additional constraints imposed by the hierarchical structure. Many Internet companies, such as popular social network and web portal sites collect web traffic and web page views (PVs) data that are naturally organized in a hierarchy. For example, web pages that are linked together often follow a hierarchical structure with the main company web site at the root and other web pages covering different aspects or functions of the company. These time series are often related in time. Modeling and forecasting of web traffic with page hierarchy is important for the core business. Intuitively, a higher volume web page category usually indicates a series of more profitable online ads placements.

In general, HTS that include related time series do not have to satisfy the equality constrains across the different hierarchical levels. For example the counts of web page views on a lower level of the web site hierarchy are not expected to be equal to the counts of web page views made at the parent level. In a website hierarchy, each time series represents the volume of daily traffics originating from the corresponding web page. Internet users might access the child web page directly without accessing the parent page, which breaks down the inclusion assumption (the observations in the parent level are strictly equal to the sum of observations of their children) and leads to the inequality constraints in the hierarchy.

In this work, we refer to HTS with equality constraints as Type I HTS and HTS with inequality constraints as Type II HTS. Even though various forecasting models are proposed for individual time series and Type I HTS, Type II HTS forecasting in web traffic domain still remain unsolved and challenging due to (1) inequality constraints which require that forecasts made at each time stamp have to incorporate hierarchical implicit inequality information to improve the prediction performance; and (2) seasonal patterns for each individual web traffic. Web traffics usually exhibit seasonal patterns and sometimes are the additive results of multiple seasonal patterns based upon human activity cycles.

In this paper, we address the above modeling issues by presenting a novel and flexible hierarchical forecasting framework to make accurate hierarchical prediction of Type II HTS. Our approach first converts the Type II HTS to Type I HTS by introducing *leak time series* and goes through multiple rounds of seasonality decomposition to remove seasonal cycles. Once done, it builds individual forecasting models on each time series without seasonality (which is referred as "trend" later in Section 4). In the final step, each individual forecasts are hierarchically combined through the hierarchy via certain combination heuristics. In summary, our approach makes the following contributions: (1) Introduce leak time series into each tree unit, which reduces the forecasting problems of Type II HTS to the forecasting problem of Type I HTS. (2) HTS with multiple seasonal patterns are automatically detected and removed and the framework uses the underlying trend as the forecasting

*This work was done when the author was in Yahoo! Labs internship.
†This work was done when the author was at Yahoo! Labs.

Conference'17, Washington, DC, USA

DOI: 10.1145/nnnnnnn.nnnnnnn

target. (3) Various individual forecasting models and hierarchical combination heuristics can be incorporated into our framework.

2 RELATED WORK

Many existing methods have been developed to forecast time series, which can be divided into two categories: univariate models and multivariate models. Univariate models can be learned directly from each individual sequence and applied to make predictions, such as polynomial regression, generalized linear models [12], local regression (LOESS) [3], Gaussian process [14], etc. Different from univariate models which treat different time series independently, multivariate time series models aim to capture and learn the dependence and interaction across many time series. Examples of such models are VAR [8], state space models [5], multi-task Gaussian process [1], etc.

Despite the fact that the various univariate and multivariate models were successfully applied into many domains, they are not suitable for the HTS forecasting. The reason is that predictions from these models usually do not satisfy the constraints imposed by HTS. By not enforcing these constraints they may suffer from the non-negligible hierarchical inconsistency errors [11]. Various solutions and heuristics have been developed in the past to tackle the forecasting problems of Type I HTS. They can be divided into the following three categories:

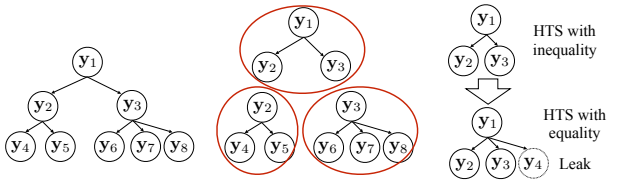
- Top-Down Heuristics. Top-Down heuristics only make forecasting for the parent nodes and disaggregate the parents' forecast into children's forecasts [15].
- Bottom-Up Heuristics. In Bottom-Up heuristics, forecasting is performed on each individual time series on the lowest level and then summed up to provide the aggregated forecasts to the corresponding levels [9, 10].
- Other Heuristics. To utilize all time series information and meanwhile make all forecasts hierarchically consistent, Hyndman et al. propose an optimal combination strategy for Type I HTS forecasting. It picks any convenient forecasting models (either univariate or multivariate models) to make the initial forecasts. After that, it infers the true low level disaggregation from the initial forecasts and aggregates it back through the hierarchy by solving a set of linear equations.

3 TERMINOLOGY & DEFINITION

Given a HTS $\mathcal{H}$, we represent each time series i as a $T \times 1$ vector $\mathbf{y}_i$. All individual time series comprise an $n \times T$ matrix $\mathbf{Y}$, where each time series corresponds to a row, explicitly, $\mathbf{Y} = [\mathbf{y}_1 \mathbf{y}_2 \dots \mathbf{y}_n]^\top$. n is the total number of time series, T is the length of each time series.

Since $\mathcal{H}$ follows a tree structure, we adopt the terminologies used in trees. A *root* is the top most node in a hierarchy. A *parent* is a time series whose observed value is equal to the sum of its children at every time stamp t, and a *child* is a time series of which a cohort compose a heavier volume time series at upper level. A *leaf* node is a child but not a parent. For example, in Figure 1(a), $\mathbf{y}_1$ is the root; $\mathbf{y}_1$, $\mathbf{y}_2$ and $\mathbf{y}_3$ are parents; $\mathbf{y}_2$, $\mathbf{y}_3$, $\mathbf{y}_4$, $\mathbf{y}_5$, $\mathbf{y}_6$, $\mathbf{y}_7$ and $\mathbf{y}_8$ are children; $\mathbf{y}_4$, $\mathbf{y}_5$, $\mathbf{y}_6$, $\mathbf{y}_7$ and $\mathbf{y}_8$ are leaves.

We denote $\mathbf{P}$ as a $p \times T$ matrix which contains all the parents in $\mathbf{Y}$; $\mathbf{L}$ is a $l \times T$ matrix which only contains all leaf nodes in $\mathbf{Y}$. We have $n = p + l$. Without loss of generality, define $\mathbf{Y} = [\mathbf{P}; \mathbf{L}]$. Similarly, $\hat{\mathbf{Y}}/\hat{\mathbf{P}}/\hat{\mathbf{L}}$ represent the forecasts from our framework for entire/parent/leaf time series and $\hat{\mathbf{Y}} = [\hat{\mathbf{P}}; \hat{\mathbf{L}}]$. Let $\mathbf{h}$ be an $n \times 1$ hierarchy indicator vector, where h_i indicates the parent of $\mathbf{y}_i$. In Figure 1(a), we have $\mathbf{h} = [0, 1, 1, 2, 2, 3, 3, 3]$. A *Unit* is the minimum subtree from $\mathcal{H}$, which is a bi-level tree structure with only one parent and at least one child. For example, we have three *Unit*s for the $\mathcal{H}$ in Figure 1(a), which are in red circles in Figure 1(b).



(a) GHTS tree structure. (b) Unit illustration. (c) Reduce to HTS with equality by using leak time series.

Figure 1: HTS notation and leak time series.

Definition 3.1. (Type I HTS) $\mathbf{Y}$ forms a Type I HTS with $\mathbf{h}$, if every parent time series $\mathbf{p}_i$ satisfies:

$$\mathbf{p}_i = \sum_{j=1}^{n} \mathbf{y}_j \cdot \mathbb{1}_{\{\mathbf{h}=i\}} \tag{1}$$

where $\mathbf{p}_i$ is the ith row in parent time series matrix $\mathbf{P}$ and $\mathbb{1}_{\{\cdot\}}$ is the indicator function, which is equal to 1 when $\{\cdot\}$ is true, otherwise it is 0.

Definition 3.2. (Type II HTS) $\mathbf{Y}$ forms a Type II HTS with $\mathbf{h}$, if there exists at least one parent time series $\mathbf{p}_i$ which does not satisfy the equality in eq.(1).

Definition 3.3. (Hierarchical Consistency) If $(\mathbf{Y}, \mathbf{h})$ satisfy eq.(1) at any time stamp, $\mathbf{Y}$ is referred to as being hierarchically consistent.

For the sake of notational brevity, we will explain our framework by using a *unit* substructure with a parent time series $(\mathbf{y}_1)$ and *n-1* time series associated with its children $(\mathbf{y}_2, \mathbf{y}_3, \cdots, \mathbf{y}_n)$. By replicating this unit substructure to form a tree, our framework can be generalized to any multi-level hierarchical tree structure.

4 THE FORECASTING FRAMEWORK

In this section, we develop a flexible forecasting framework for Type II HTS. It includes four steps: (1) Type II HTS reduction; (2) Seasonality decomposition; (3) Trend forecasting; and (4) Hierarchical combination. Details are as follows.

4.1 Step1: Type II HTS Reduction

Although HTS modeling and forecasting are crucial to Internet companies, the modeling difficulties for Type I HTS and Type II HTS vary. In Type I HTS, the equality constraints implicitly provide strict bounds for observations from each individual time series. However, because there are no explicit constraints to bound the forecasts for each time series in Type II HTS modeling, predicted values can be any real number. Furthermore, in Type II HTS modeling, the hierarchy information is hard to incorporate in the modeling and forecasting phases.

Our framework provides a simple solution to handle Type II HTS modeling problem by reduction. We reduce Type II HTS to Type I HTS by introducing the leak time series on the child level for each unit in the hierarchical tree. The leak time series varies its value at every time stamp to ensure the hierarchical consistency, which reduces Type II HTS to Type I HTS. This reduction is illustrated in Figure 1(c).

4.2 Step2: Seasonal Decomposition

Time series may contain multiple seasonal cycles of different lengths, which is quite common in the real life. In Figure 2, we show a two-year long time series of daily PVs for a web portal site homepage. As we can see, the time series shows both a weekly and a yearly pattern. Briefly, Internet visitors tend to visit the homepage more frequently during Monday and Tuesday and fewer people come to the site during the weekend. Considering longer horizon, the time series also exhibits a yearly pattern with dramatic PV drops during the holiday seasons, such as Christmas holiday.

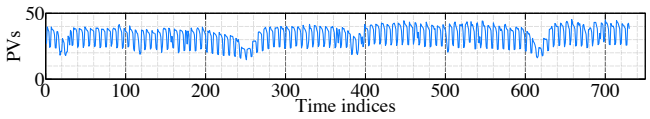


Figure 2: Daily PVs for the homepage in Italy over two years.

Multiple nested seasonal patterns pose a big challenge for time series modeling since the nested cycles easily deceive traditional time series models and make them fail to find the underlying changes in the time series [4]. In our framework, we iteratively extract seasonal pattern by applying "STL Decomposition", which is a filtering procedure for decomposing a time series into *seasonality*, *trend* and *irregularity* components [2]. To illustrate this, we apply the seasonal decomposition to the daily PV time series in Figure 2. The resulting seasonality, trend and irregularity components are shown in Figure 3. As we can see, (1) the weekly pattern is extracted automatically; (2) the yearly pattern becomes obvious after the removal of the weekly pattern; and (3) random noise (irregularity component) are removed from the original time series which leads to a smoother trend series.

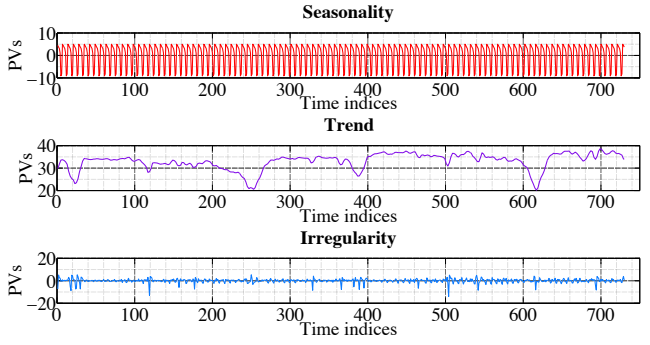


Figure 3: STL decomposition of Italy homepage daily PVs.

In this step, we keep iterating the seasonality decomposition until all the seasonal patterns are removed. Due to the space limit, mathematical details of the "STL Decomposition" approach are omitted and interested readers can find them in [2].

4.3 Step3: Trend Forecasting

After the removal of seasonal and irregular (residual) components, a smoother trend time series remains. This time series becomes our forecasting target. Generally speaking, any forecasting model from either univariate or multivariate models in Section 2 can be incorporated into our framework to make trend predictions. Due to space limit, we choose to model the trend series by using autoregressive integrated moving average (ARIMA) [7] as an illustration example.

Autoregressive integrated moving average. ARIMA is widely used in univariate time series forecasting. Let $y_{i,t}$ be the observation from time series i at time stamp t and $\Delta y_{i,t} = y_{i,t} - y_{i,t-1}$, $\Delta^2 y_{i,t} = \Delta(\Delta y_{i,t})$, and so on and an ARIMA$(r, s, q)$ model is defined as follows:

$$y^*_{i,t} = \phi_1 y^*_{i,t} + \cdots + \phi_r y^*_{i,t-r} + \eta_t + \theta_1 \eta_{t-1} + \cdots + \theta_q \eta_{t-q}$$

where $y^*_{i,t} = \Delta^s y_{i,t}$ and $\{\eta_t\}$ is a serially independent series of $\mathcal{N}(0, \sigma^2_\eta)$ disturbances. Parameters r, s, and q are non-negative integers that refer to the order of the autoregressive, integrated, and moving average parts of the model respectively.

Once the trend forecasting model is selected, our *HF* framework will directly train the model based on the trend series, which is obtained from previous *Seasonal Decomposition* step and make the predictions. Seasonality (represented in terms of additive constants) is added back to the trend predictions to form the final forecast.

4.4 Step4: Hierarchical Combination

In order to ensure the Type I HTS hierarchical consistency, various heuristics can be designed to adjust the initial forecasts (forecasts from the previous *Trend Forecasting* step). These heuristics were reviewed in Section 2. All of them can be applied to produce a hierarchically consistent forecasts as follows.

Top-Down with historical proportion (TD). We predict the parent time series directly and split parent forecasts into disaggregations based on children time series' historical proportions.

Bottom-Up (BU). We predict children time series directly using the forecasts and aggregate the children's forecasts for the parent time series predictions.

Optimal combination forecasts (OPT). We adjust all the forecasts from the previous step (Step3) according to the optimal combination forecast strategy in [6]. OPT assumes that any forecasting sequence from the previous step is a linear combination of the true means of the leaf time series ($\tilde{\mathbf{L}}$) in the hierarchy and the entire forecasting from *Trend Forecasting* step $\hat{\mathbf{Y}}$ can be expressed concisely as $\hat{\mathbf{Y}} = \Omega \cdot \tilde{\mathbf{L}} + \epsilon_{\mathbf{Y}} = \Omega \cdot (\tilde{\mathbf{L}} + \epsilon_{\mathbf{L}})$ where $\hat{\mathbf{L}} = \tilde{\mathbf{L}} + \epsilon_{\mathbf{L}}$ and Ω is the $n \times l$ summing matrix and each row of Ω, noted as Ω_i, defines the linear combination coefficients for the corresponding time series i. Ω on $\mathcal{H}$ is made of two parts: (1) parent linear combination coefficients matrix $\Omega_{\mathbf{P}}$; (2) the $l \times l$ identity matrix $\mathbf{I}_{l \times l}$. $\Omega = [\Omega_{\mathbf{P}}; \mathbf{I}_{l \times l}]$. $\Omega_{\mathbf{P}}$ is built in a bottom-up fashion row by row, where $\Omega_{P_i} = \sum_{j=1}^{n} \Omega_j \cdot \mathbb{1}_{\{\mathbf{h}_j = i\}}$. For instance, the summing matrix for HTS in Figure 1(a) is $\Omega = [1\ 1\ 1\ 1\ 1; 1\ 1\ 1\ 0\ 0; 0\ 0\ 0\ 1\ 1; \mathbf{I}_{5\times5}]$.

The OPT heuristic adjust the initial forecast by using the hierarchical consistency projection operator $\mathcal{P}_{OPT}$ defined as $\mathcal{P}_{OPT}(\hat{\mathbf{Y}}, \Omega) = \Omega(\Omega^\top \Omega)^{-1} \Omega^\top \hat{\mathbf{Y}}$.

5 EXPERIMENTS

We evaluate the performance of our proposed method on a real-world web traffic dataset from a popular web portal site. Our dataset *MEDIA* is a Type II HTS dataset with a website hierarchy. *MEDIA* records daily PV traffic to the "media" homepage. The root "media" page has 6 children that represent the daily PVs from sports, auto, finance, news, and celebrity web pages. The "sports" page has two children in the next level, which represents the PV volumes from different channels (desktop or mobile). It covers the period of 01/01/2013 to 15/03/2015. *MEDIA* exhibits weekly and yearly seasonal patterns. In this experiment, we separate the last month's data of *MEDIA* as our test data and use all the previous data for training purpose. We would also like to note that the hyper parameters, such as the model orders in ARIMA models (r, s and q) used in our methods are selected by the internal cross validation approach while optimizing models' forecasting performances.

We compare our framework to representatives of the different approaches: (1) Seasonal ARIMA Model (SARIMA). The extended ARIMA model specializes in modeling seasonal time series[7]; (2) Gaussian process with periodic kernels (GP). [14]; and (3) Linear dynamical system (LDS). LDS is a widely used forecasting model from dynamic Bayesian network family [13]. We denote our approach as "HF-ARIMA" or "HF-A" for short.

We evaluate and compare the performance of the different methods by calculating the average mean absolute percentage Error (Avg-MAPE) [11]. Avg-MAPE measures the forecasting accuracy, which is defined as Avg-MAPE $= \frac{1}{T}\sum_{t=1}^{T}\cdot|1-\hat{y}_{it}/y_{it}| \times 100\%$, where $\hat{y}_{it}$ is the forecast from our framework and y_{it} is the true forecast for time series i at time stamp t.

5.1 Discussion

First, we study the influence of different combination heuristics. We apply different hierarchical combination heuristics (TD, BU and OPT) to the forecasts generated from our framework. The Avg-MAPE results are reported in Table 1. From Table 1, we can see that (1) OPT approach has the most robust performance and in the majority of time series forecasting tasks, it gives the lowest Avg-MAPE; (2) TD approach always ends up with the worst performance due to the fact that only the parent time series are modeled, and all children are disaggregated proportionally to the forecasted parent value. In TD approach, all children time series information is discarded, which leads to a massive information loss, thus the inferior performance.

Table 1: HF-ARIMA on *MEDIA* dataset with different hierarchical combination heuristics.

Method	Media	Sports	Sports_Mobile	Sports_Desktop	Auto	Finance	News	Celebrity
HF-A-TD	11.20	28.90	26.31	64.08	26.15	55.52	14.74	81.98
HF-A-BU	**11.20**	18.86	14.51	**32.39**	39.73	9.75	**12.96**	33.77
HF-A-OPT	**11.20**	**18.84**	**14.48**	32.40	39.77	**9.74**	**12.96**	33.78

Next, we compare our framework with other forecasting baselines. Here we choose to use OPT heuristic to ensure the hierarchical consistent forecast, and the results are reported in Table 2. As we can see, the forecasting methods from the our framework almost always dominate all the forecasting tasks. Even in some rare cases where other methods achieve the best Avg-MAPE, forecasting errors from our framework are very close to the smallest errors. On the contrary, baseline methods usually could not maintain stable forecasting performance, and their forecasts tend to diverge under imprecise settings.

Table 2: Avg-MAPE results on *MEDIA* with OPT heuristic.

Method	Media	Sports	Sports_Mobile	Sports_Desktop	Auto	Finance	News	Celebrity
SARIMA	62.81	71.92	81.98	117.02	654.08	12.80	10.86	46.89
GP	55.02	234.86	37.60	188.89	162.23	101.73	118.79	320.29
LDS	29.86	59.88	18.20	101.86	**31.96**	15.40	**10.62**	200.80
HF-A-OPT	**11.20**	**18.84**	**14.48**	**32.40**	39.77	**9.74**	12.96	**33.78**

In summary, HF-ARIMA with OPT heuristics seems to be most appropriate for web time series traffic forecasting. We believe this is because (1) different PV traffics have very different seasonalities across parent and children, and within our framework, each seasonality could be precisely identified; (2) our framework is able to remove noises and extract smooth trends from the original time series, which unburden the difficulties of trend modeling; (3) OPT heuristics utilizes the information from individual time series forecasts as much as possible, which is much more effective compared to TD (simple propositional disaggregation) and BU (naive summation).

6 CONCLUSION

In this paper, we propose a new flexible framework for hierarchical time series forecasting. Our framework is able to make accurate forecasting with both the Type I and Type II HTS with multiple seasonal patterns by (1) reducing Type II HTS to Type I HTS by using the leak time series; (2) iteratively decomposing HTS into seasonality and trend; (3) building accurate trend forecasting models; and (4) applying combination heuristics to make hierarchically consistent forecasts. Experiments on two real-world datasets demonstrate (1) effect of the different combination heuristics on the final forecasts; and (2) our framework outperforming other state-of-the-art HTS forecasting approaches. In the future, we plan to extend this work to other types of probabilistic models that may have a better model interpretation.

REFERENCES

[1] Edwin Bonilla, Kian Ming Chai, and Christopher Williams. 2008. Multi-task Gaussian process prediction. (2008).

[2] Robert B Cleveland, William S Cleveland, Jean E McRae, and Irma Terpenning. 1990. STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics* 6, 1 (1990), 3–73.

[3] William S Cleveland. 1979. Robust locally weighted regression and smoothing scatterplots. *Journal of the American statistical association* 74, 368 (1979), 829–836.

[4] Alysha M De Livera, Rob J Hyndman, and Ralph D Snyder. 2011. Forecasting time series with complex seasonal patterns using exponential smoothing. *J. Amer. Statist. Assoc.* 106, 496 (2011), 1513–1527.

[5] James D Hamilton. 1994. State-space models. *Handbook of econometrics* 4 (1994), 3039–3080.

[6] Rob J Hyndman, Roman A Ahmed, George Athanasopoulos, and Han Lin Shang. 2011. Optimal combination forecasts for hierarchical time series. *Computational Statistics & Data Analysis* 55, 9 (2011), 2579–2589.

[7] G Janacek. 2010. Time series analysis forecasting and control. *Journal of Time Series Analysis* 31, 4 (2010), 303–303.

[8] Soren Johansen. 1995. Likelihood-based inference in cointegrated vector autoregressive models. *OUP Catalogue* (1995).

[9] Kenneth B Kahn. 1998. Revisiting top-down versus bottom-up forecasting. *The Journal of Business Forecasting Methods & Systems* 17, 2 (1998), 14.

[10] Robert Kohn. 1982. When is an aggregate of a time series efficiently forecast by its past? *Journal of Econometrics* 18, 3 (1982), 337–349.

[11] Zitao Liu, Yan Yan, Jian Yang, and Milos Hauskrecht. 2015. Missing Value Estimation for Hierarchical Time Series: A Study of Hierarchical Web Traffic. In *ICDM*. 895–900.
[12] Peter McCullagh and John A Nelder. 1989. Generalized linear models. (1989).
[13] Kevin Patrick Murphy. *Dynamic bayesian networks: representation, inference and learning*. Ph.D. Dissertation.
[14] Carl Edward Rasmussen. 2006. Gaussian processes for machine learning. (2006).
[15] Handik Widiarta, S Viswanathan, and Rajesh Piplani. 2008. Forecasting item-level demands: an analytical evaluation of top–down versus bottom–up forecasting in a production-planning framework. *IMA Journal of Management Mathematics* 19, 2 (2008), 207–218.