Disjoint-Support Factors and Seasonality Estimation in E-Commerce

Abhay Jha

Facebook, Inc., Menlo Park CA, USA[*]
abhaykj@fb.com

Abstract. Successful inventory management in retail entails accurate demand forecasts for many weeks/months ahead. Forecasting models use *seasonality*: recurring pattern of sales every year, to make this forecast. In e-commerce setting, where the catalog of items is much larger than brick and mortar stores and hence includes a lot of items with short history, it is infeasible to compute seasonality for items individually. It is customary in these cases to use ideas from factor analysis and express seasonality by a few factors/basis vectors computed together for an entire assortment of related items. In this paper, we demonstrate the effectiveness of choosing vectors with disjoint support as basis for seasonality when dealing with a large number of short time-series. We give theoretical results on computation of disjoint support factors that extend the state of the art, and also discuss temporal regularization necessary to make it work on walmart e-commerce dataset. Our experiments demonstrate a marked improvement in forecast accuracy for items with short history.

1 Introduction

Seasonality refers to patterns in a time-series that repeat themselves every season. For example, retail sales always increase in November, unemployment drops in December, temperature increases in summer. In general, one is interested in finding the smooth periodic pattern underlying a long univariate time-series which has data for many past seasons. This reduces to some form of regression of observation on the season, for e.g., day/week, as exemplified in a lot of time-series literature [1–3].

In this paper, we will focus on finding the weekly seasonality of sales on an annual basis. We focus on e-commerce, which is a decidedly different and arguably more challenging task, because the assortment of items is larger and more dynamic– this implies there is a large number of time-series and most of them do not have enough data for even one year. This make the traditional approach of regression infeasible, since we cannot estimate a 52 week seasonality from, say only 6 weeks of sales. The problem in this domain is more suited to factor analysis and matrix factorization techniques, which have been successfully used for imputation in other scenarios with a lot of missing data [4]. In this

[*] Work done while the author was at @WalmartLabs

approach, one computes instead a few orthogonal basis vectors, called seasonal basis for an entire category of related items. Figure 1 illustrates the seasonal basis of a certain group of items when computed on the online sales data over 52 weeks of the year. Seasonality for an item can be evaluated with a regression, generally by a time-series forecasting model with time varying coefficients, on the seasonal basis. We illustrate a simple forecasting model that incorporates seasonal basis in (1). However, this regression can lead to unreliable results for two reasons. First, in the span of a short time-series, individual seasonal basis might not be orthogonal. For e.g., in Fig. 1, basis 2 and 3 from PCA have very similar curve from week 20 to 35 and same with SPCA for weeks 35 to 52, which makes it impossible to disambiguate between them if a time-series only had data for those weeks. One solution is to work with fewer basis; but unless one always works with one basis, there is no guarantee that they would be orthogonal for every segment. This is a big issue when a vast majority of items being forecasted don't even have a year of data.

Another, a more intuitive problem, is that not all parts of the year are related to each other. For an item, sales in February may have no relation to sales in September, but may be related to sales in December. Hence, we should not be modifying forecasts for the entire year based on the sales during a part, which is what happens in general. Fortunately, both problems lead to one solution. To solve the first problem, we enforce a stricter notion of orthogonality where every segment of two vectors is orthogonal; it can be shown to be equivalent to them having Disjoint Support(DS). Figure 1 illustrates DS-basis. They solve the second problem as well by segmenting the year into different weeks which exhibit a distinct behavior. However, with disjoint supports there is only one curve to model the variation during any part of the year, which is not always true when one considers a large group of possibly unrelated items. So, one way of viewing DS-basis is as a strong regularizer imposed on a group of items which forces their sales to follow one curve during the seasonal events. This is not recommended for the entire catalog together, but for groups of related items in the catalog hierarchy.

In this paper, we will study how to compute DS-basis, both with theoretical results, and practical lessons learned in applying it at walmart e-commerce. We show that DS-basis for a low rank matrix can be computed in polynomial time. Our proof relies on bounding the number of regions of a a low rank hyperplane arrangement. For general matrices, we show that the problem is NP-hard, with a reduction from graph coloring. We also give a constant factor approximation algorithm, and prove hardness of approximation results.

When applying this technique at Walmart E-Commerce, we observed multiple anomalies in the basis computed, compared to our domain knowledge. This is because real world datasets are noisy and there are many factors that lead to variation in sales that cannot be accounted for. We propose certain temporal regularizations that can overcome this noise, by exploiting the fact that our data is a time-series. Computing the basis with these regularization entails learning in Switching State Space Models which is often done with moment-matching

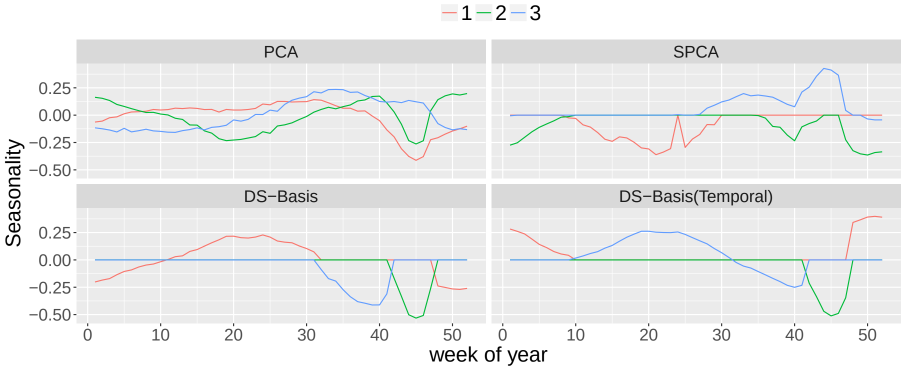

Fig. 1. Seasonal factors computed with different methods on a group of items from walmart-ecommerce. Notice how basis 2 and 3 from PCA have very similar curve from week 20 to 35 and same with SPCA for weeks 35 to 52. This makes it impossible to disambiguate between the two basis if a time-series only had data for those weeks. This leads to unreliable estimate of seasonality and hence unreliable forecasts. Unless one works with only one basis, this problem is inevitable; hence the notion of disjoint support basis that lead to orthogonality for every segment. *DS*-basis only have one non-zero component at a time– the curves sometimes seem to overlap as one basis goes from zero to non-zero and another from non-zero to zero.

Kalman Filters [5]. We propose an alternative faster approach that leverages forward-backward algorithm for estimation in HMM, to achieve the same accuracy, in less execution time.

Our experiments demonstrate that forecast accuracy is markedly improved for items with short history. Further empirical evaluations are done on a synthetic dataset for a more detailed comparison. This paper is organized as follows: Sect. 2 gives some background and discusses the related work, and we describe our approach along with the computational complexity of the problem in Sect. 3, and Sect. 4, which focuses on a more general problem with temporal regularizations. Section 5 has the empirical evaluation of our approach conducted on walmart e-commerce, and a synthetic data.

2 Background and Related Work

Related Work The general approach of estimating seasonality is by decomposing the time-series into mean, trend and seasonal components, see [6] for an example. The seasonal component can be modeled as a cyclic/periodic component in the form of a triginometric series. However, because of leap years, seasonality in our setting is not the same as periodicity. However, it can still be computed by a regression of observation with the week number– but as we already pointed out this approach does not work for short series. One could still use hierarchical regressions [7] commonly used in panel data– in this approach we would use item catalog to specify a hierarchy of items, however this approach does not scale well to large datasets we use. This is inherent to the method itself because of computations involving large covariance matrices. Also, generally some sort of clustering is needed before applying these methods as described in [8].

The application of forecasting to settings like ours is relatively new, but the idea of seasonal factors is not new and has been investigated by others as well. For e.g., [9, 10], explore Non-Negative Matrix Factorization(NMF), and Principal Component Analysis(PCA) respectively. In this paper, we are focused not on the particulars of whether factors be non-negative, or sparse or smooth; instead we are proposing that they have disjoint support which is an orthogonal idea that can be used along with each of these approaches. We build upon PCA since it is the most common way of estimating factors.

Notation Given a vector v, we denote coordinate i by v_i. For a matrix M, we denote row i and column j with $M[i, j]$; column i with m_i. We say vectors u, v have disjoint support iff $\forall i, u_i v_i = 0$. The reason we are interested in them is because they ensure orthogonality of arbitrary segments of u, v. For a natural number n, $[n]$ denotes the set $\{1, 2, \ldots, n\}$.

Forecasting with Seasonal Basis Since this paper is about computing seasonality, we won't delve into the forecasting models, but we do want to illustrate how seasonality is incorporated in forecasting to motivate the problem. The following is a simple univariate local-level model which has a *mean* component μ and a seasonality component that is expressed by seasonal basis that form rows of H.

$$\begin{aligned} y_t &= \mu_t + h_{s(t)}^T \alpha_t + \epsilon_t & \epsilon_t &\sim N(0, \sigma^2), \\ \mu_t &= \mu_{t-1} + \eta_t & \eta_t &\sim N(0, \lambda_\mu \sigma^2), \\ \alpha_t &= \alpha_{t-1} + \omega_t & \omega_t &\sim N(0, \lambda_\omega \sigma^2 \mathbf{I}_k) \end{aligned} \tag{1}$$

where $s(t)$ denotes the season at time t. For e.g., if we are making weekly forecasts it will be between 1 to 52, for daily forecasts it would vary from 1 to 365. One could add more components to the model like trend, and include price and calendar effects. Furthermore, this could be generalized to a multivariate model.

Problem Statement Let Y be an $n \times p$ sales matrix, where p is the number of seasons, for e.g. $p = 52$ in Fig. 1. We will assume that rows of Y are centered to take out the effect of mean. Also, note that Y can have missing values. Our goal in this paper is to express Y as WH, where W is an $n \times k$ matrix of basis coefficients, and H is a $k \times p$ matrix whose rows have disjoint support and $HH^T = I$, that minimize $\|Y - WH\|_F^2$, which is same as maximizing $Tr\left(HY^TYH^T\right)$. Note that the constraint $HH^T = I$ is just for uniqueness; we could also enforce $W^TW = I$ instead– depending on the algorithm one constraint is preferred over the other. We will extend the problem further by adding some temporal constraints on the rows of H in Sect. 4. k is typically a small number, and hence would be assumed to be a bounded constant when stating complexity results throughout this paper.

```
Input: $Y_{n\times p} = U_{n\times r} V_{r\times p}$, #factors $k$
Output: $W$, $H$
$Z_{r\times k} \leftarrow$ variables from a $rk$-D space
$S \leftarrow \bigcup_{1\leq i\leq p} \bigcup_{1\leq j_1<j_2\leq k} \{v_i^T z_{j_1} \pm v_i^T z_{j_2} = 0\}$
$r(S) \leftarrow$ regions of arrangement $S$
$opt \leftarrow 0$
for regions $\nabla \in r(S)$ do
    $Support(i) \leftarrow \texttt{argmax}_j \left(v_i^T z_j\right)^2, \forall i \in [p]$
    $M_j \leftarrow$ columns of $Y$ with support $j, \forall j \in [k]$
    $currOpt \leftarrow \sum_{i=1}^{k} \sigma_1^2(M_i)$
    if $currOpt > opt$ then
        $opt \leftarrow currOpt$
        $H \leftarrow$ right singular vectors of $M_i, i = 1..k$
    end
end
$W \leftarrow YH^T$
return $W$, $H$
```

Algorithm 1: Computing *DS*-basis for a low rank matrix

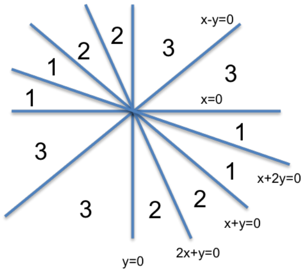

Fig. 2. Consider functions 1.x^2, 2.y^2, 3. $(x+y)^2$. The above arrangement of lines partitions 2D-space into regions which are annotated with 1/2/3 according to which function is maximum in that region. The lines are just $f \pm g$ for each pair of functions f^2, g^2.

3 Computing *DS*-Basis

We first discuss the low rank case and propose a polynomial time algorithm. The algorithm can be used in general too by applying it on a low rank projection. We then discuss the results for general matrices including NP-hardness and approximation results. Finally, we show how these results can be extended if basis need to be sparse.

3.1 *DS*-Basis for Low Rank Matrices

We first reformulate the problem from Sect. 2 into a form that depends only on W, and not on the basis H. W.l.o.g, we assume $\|w_i\|_2 = 1, \forall i \in [k]$. Now, note that if the i^{th}-support is basis j, then $H_{j,i} = w_j^T y_i$. Hence, it follows that the optimal W can be found by maximizing:

$$\sum_{i=1}^{p} \texttt{max}\left(\left(w_1^T y_i\right)^2, \ldots, \left(w_k^T y_i\right)^2\right)$$

s.t. $\|w_i\|_2 = 1, i \in [k]$. Now we use the fact that that Y is low rank to express it as $Y = UV$, and replace W, nk variables, with variables $Z = W^T U$, only kr

variables. We can then reformulate the objective as:

$$\sum_{i=1}^{p} \texttt{max} \left(\left(z_1^T v_i \right)^2, \ldots, \left(z_k^T v_i \right)^2 \right) \tag{2}$$

maximization of a low-rank convex function over the unit sphere. A Polynomial Time Approximation Algorithm(PTAS) for this problem is possible by iterating over the unit sphere for Z, by discretizing it into grids of small size, and using the property that the change in objective is bounded by ϵ^2, for a perturbation in Z of ϵ. This has been already discovered in [11], and similar approach has also been used in [12] to maximize a class of quasi-convex functions. But in neither of these cases, is an algorithm with polynomial running time independent of the error ϵ still known. In this paper, we present such an approach, that can also extend the result in [11] from PTAS to PTIME, and we hope can be extended to more low rank convex maximization problems as in [12].

Algorithm 1 presents the algorithm for computing *DS*-basis of a low rank matrix. We restate that k and the rank of Y are assumed to be small constants in this section. Formally:

Theorem 1 (Computing *DS*-basis of a low rank matrix is in PTIME). *Given a matrix Y of rank r, we can compute DS-basis H of Y in time $O\left(p^{rk+4}\right)$*

3.2 *DS*-Basis for Arbitrary Matrices

Once we venture beyond low-rank matrices to arbitrary matrices, the problem becomes NP-hard as we prove below.

Theorem 2 (Computing a *DS*-basis of even constant size is NP-hard). *Finding a DS-basis H that maximizes $Tr\left(HY^TYH^T\right)$ s.t. $HH^T = I_k$ is NP-hard for any fixed $k \geq 3$.*

In general, it would be good to know how close one could approximately solve the problem for a general matrix. We give an incomplete answer by proving both lower and upper bounds on the optimal hardness of approximation. It would be great if the two matched so we knew how close an approximation is possible in polynomial time, but we leave that as an open problem.

Theorem 3 (Approximating a *DS*-basis of size k). *Let $\texttt{max}_H Tr(HY^TYH^T)$ be opt*, where H is a DS-basis and $HH^T = I_k$. Then opt*, can be approximated to a ratio $1/k$ in PTIME. Furthermore, unless P=NP, it cannot be approximated to a ratio better than $1 - {}^1\!/_p$ in PTIME.*

Implications for Sparse PCA. Algorithm 1 is very general in its scope, in that it first shows that there are only a polynomial and not exponential number of possible supports one needs to consider when looking for disjoint support of a low rank matrix. Of course, given the support one still needs to find the basis, which reduces to finding the dominant eigenvector of a low rank matrix. The framework extends to solving for principal components with a particular constraint as well, so long as the second stage is still tractable. In case of sparse pca, for instance, one can find the principal component of a low rank matrix with either l_0/l_1 constraint in polynomial time [13, 14]. Hence the tractability results extend to these cases as well.

4 Adding Temporal Regularization

As can be seen from Fig. 1, *DS*-basis sometimes look counterintuitive, when the non-zero component switches between consecutive weeks of year for a short period of time. In general, we expect the year to be divided into contiguous segments of weeks that form the support for a basis, and non-contiguous supports are implausible. But in a noisy real-world dataset like ours, it can be hard to get to the optimal solution due to the presence of multiple outliers and other noise. To counter the noise, we enforce this domain knowledge via a prior/regularizer over the simple gaussian factor analysis approach as follows. We model the support using a Hidden Markov Model(HMM) that encourages consecutive time-periods to have similar support. However, we also have to account for the fact that our data is a time-series. This means we expect our basis curves to be smooth and not change too much from one time point to another.

$$\begin{aligned} y_t &= w_{x_t} H_{x_t,t} + \epsilon_t \qquad \epsilon_t \sim N\left(0, \sigma^2 \mathbf{I}\right) \\ \mathbf{Pr}\left(x_t | x_{t-1}\right) &= \rho \mathbf{1}_{x_t = x_{t-1}} + {}^{(1-\rho)}/_{(k-1)} \mathbf{1}_{x_t \neq x_{t-1}} \\ h_t &= h_{t-1} + \eta_t \qquad \eta_t \sim N\left(0, \lambda \sigma^2 \mathbf{I}\right) \end{aligned} \tag{3}$$

These two regularization work against each other. Regularization on the support indicator x_t tries to put consecutive seasons in the same support– this is tuned with the parameter ρ, if it is ${}^1/_k$, supports can change arbitrarily between time points, while if $\rho = 1$, consecutive weeks must have the same support leading to only one non-zero basis. There is also a penalty on the difference in basis h between consecutive weeks, controlled by parameter λ– higher λ means lower penalty, while $\lambda = 0$ forces h to be constant. The seasonality and segmentation achieved with these regularizations look more natural, and as we will show in Sect. 5, lead to better forecast accuracy as well. However, because the consecutive supports are correlated, an approach like Algo. 1 is no longer applicable.

Equation 3 is a special case of *Switching State Space Models*(SSSM) which combine ideas from HMM and State Space Models(SSM) to allow for both discrete and continuous hidden states. Unlike SSM, computing the distribution of H given W has been recognized as intractable in SSSM [15]. The hardness of computing posterior state distribution stems from the fact that at each time point, we have k possibilities corresponding to the values of x_t. The final posterior thus is a mixture of k^p gaussians. Various approximations have been used in the literature to deal with this intractability. The most common is to modify the kalman filter by merging the k gaussians into 1 gaussian at each step [5, 16, 17]: the resulting filter is called GPB(1). This leads to a natural alternating minimization scheme, which we call AM-GPB(1), summarized in Algo. 2: compute

H given W using GPB(1), and W given H using regression. Time complexity per iteration can be shown to be $O\left(npk^4\right)$, dominated by the time for GPB(1).

However, GPB(1)-smoothing is expensive and the execution cost builds up because of the repeated calls involved with the alternating minimization involved. We also pursue an alternative way in which we put more emphasis on finding states x instead. Observe that given support x, we can find basis i as the first eigenvector of $Y^{(i)T}Y^{(i)} + {}^{L}\!/\!_{\lambda}$, where $Y^{(i)}$ is the matrix with columns of support i from Y, and L is a tridiagonal matrix with 2 on diagonal, except the first and last, and -1 off-diagonal. Also, once we know H, W is just YH^T. Now, we use W to find x using the forward backward algorithm for state estimation in HMM. Note that this step completely ignores H, and just finds optimal x for the given W. In other words, while GPB(1) smoothing focuses more on estimating H, this approach puts more emphasis on x. The time complexity per iteration now is $O\left(npk + p^3\right)$. We call it AM-HMM, summarized in Algo. 3, and it also leads to a faster execution time as we will demonstrate empirically.

5 Empirical Evaluation

In this section, we will look at the impact of *DS*-basis on forecast accuracy in a real-world dataset, and also also explore the robustness and performance of the algorithms proposed in this paper on a synthetic dataset. Our implementation is in C++ and R, and experiments are conducted on a MacBook Pro with 16GB memory and 2.5 GHz Intel i7 processor.

5.1 E-Commerce Data

In this section, we use sales data from Walmart E-Commerce. 20 groups of items from different sections of the catalog are selected, with sizes varying from about 2K to 10K, for a total of around 50K items. We should point out that these groups were not manually selected; they are actual groups of items assigned to a particular category in the catalogue. In that sense, the items they contain are representative of an e-commerce assortment. We will compare the forecasts from local level model in (1) with $k = 3$, $\lambda_\mu = \lambda_\omega = 0.1$. For forecasts, we choose six different weeks of year distributed throughout year. For each week, we forecasted six weeks ahead. Our benchmark for comparison are seasonal factors generated using PCA. To compare how a new forecast f compares to benchmark g, we look at the metric of percentage improvement offered by f over g: ${}^{|f-s|-|g-s|}\!/\!_{|f-s|}$, where s is the sales. We compute *DS*-basis for each group using Algo. 1[1], and *DS*-basis(temporal) with temporal regularization is computed using AM-HMM.

Figure 3 shows that there is a stark difference in comparison when it comes to items with less than a year of history and items with long history, with median improvement of 20-30% possible with *DS*-basis. This is in accordance with the

[1] We don't explore the full search space but use randomization to run within a time budget

```
Input: Sales Matrix $Y(n \times p)$,
       initial $W = W^0$, parameters
       $k, \sigma, \rho, \lambda$
Output: $W$, $H$
$W \leftarrow W^0$
while $H$ has not converged do
    // Compute $H, x$ given $W$
    Compute states $x_t, h_t$ with
      GPB(1) smoothing [5,17]
    Normalize each row of $H$ to
      norm 1
    // Compute $W$ given $H$
    $W \leftarrow YH^T$
end
return $W$,$H$
```

Algorithm 2: AM-GPB(1) algorithm to compute *DS*-basis

```
Input: Sales Matrix $Y(n \times p)$,
       initial $W = W^0$,parameters
       $k, \sigma, \rho, \lambda$
Output: $W$, $H$
$W \leftarrow W^0$
while $H$ has not converged do
    // Compute $x$ given $W$
    Compute $x_t$ using Viterbi
      algorithm
    // Compute $H$ given $x$
    for $i \in [k]$ do
        $s \leftarrow \{j \mid x_j = i\}$ // columns
          with support $i$

        $Y_s \leftarrow$ matrix with columns
          $y_j \forall j \in s$

        $L \leftarrow \begin{bmatrix} 1 & -1 & 0 & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \ddots & \vdots \\ 0 & -1 & 2 & -1 & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & -1 & 2 & -1 \\ 0 & \dots & 0 & 0 & -1 & 1 \end{bmatrix}$

        $h_i \leftarrow$ first eigenvector of
        $Y_s^T Y + L/\lambda$
    end
    // Compute $W$ given $H$
    $W \leftarrow YH^T$
end
return $W$,$H$
```

Algorithm 3: AM-HMM algorithm to compute *DS*-basis

argument made in Sect. 1 that having orthogonal basis is not sufficient when the time-series involved are short, since it can be hard to disambiguate between different factors in a short time-span. But not only do we see improvements for short time-series, we don't experience any penalty for long time-series when using *DS*-basis(temporal) which is encouraging since it means the approach can be deployed for all items and not restricted to short series.

Figure 4 describes in detail how the improvement offered by *DS*-basis(temporal) varies with length of history a time-series has. We only plot the average improvement for items with given weeks of history to minimize the clutter of the graph resulting from too many points. Figure 4 shows, if we ignore the beginning, till

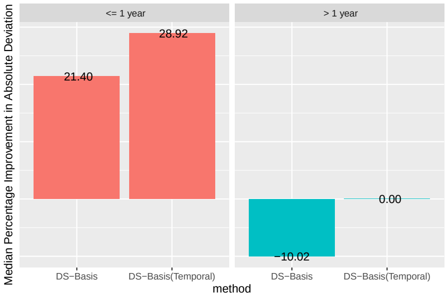

Fig. 3. Median Percent Improvement in error for items with less than or more than one year of sales history by using *DS*-basis over principal components. Improvement is ${|f-s|-|g-s|}/{|f-s|}$, where s is sales, f, g are forecasts using seasonality from PCA and *DS*-basis. *DS*-basis was computed using Algo. 1, and *DS*-basis(temporal) with temporal regularization is computed using Algo. 3

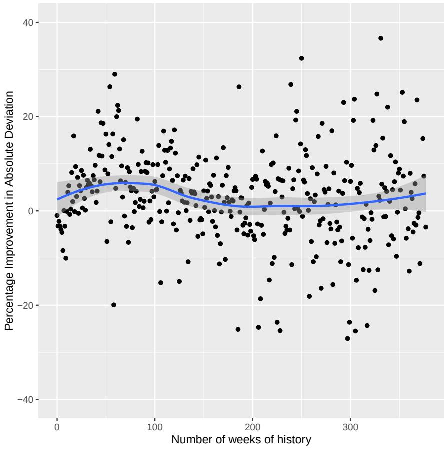

Fig. 4. Average Percent Improvement in error for items with a certain week of sales data; the shaded region shows the 95% confidence interval. This shows significant improvements for items with short time-series. Improvement is ${|f-s|-|g-s|}/{|f-s|}$, where s is sales, f, g are forecasts using seasonality from PCA and *DS*-basis computed by AM-HMM respectively.

say 10 weeks, there is a clear and marked improvement for items with less than 60 weeks of sales, often about 10-25%. For items with less than 10 weeks of history, initialization is the dominating factor, and performance is very volatile. From 50 to 150, most of the times improvement is positive, but after 150 weeks, there is no significant improvement.

5.2 Synthetic Data

In this section, we will evaluate our algorithms for computing *DS*-basis, and see if they are effective in finding the underlying basis and observation in the presence of noise and outliers, assuming that the underlying basis does have disjoint support. For this, given $0 \leq f \leq 1$, we generate a matrix M of dimension 1000×52 as $M = WH + \epsilon + \mu$, where W is 1000×3 matrix of $\mathcal{N}(0,1)$, and H is a 3×52 smooth disjoint support factor where factors vary from one time-point to another by $\mathcal{N}(0, 0.1)$. ϵ is $\mathcal{N}(0,1)$ error and μ is outlier noise: with probability f it is $\mathcal{N}(0, 10)$, else it is zero. Now to simulate the missing data, we divide rows of M into 50 groups and from each group remove the first $0, 1, \ldots, 49$ entries. Note that M is then about 50% sparse, but in a stair-case fashion since we assume the data is time-series and hence the missing data is at the beginning and not at

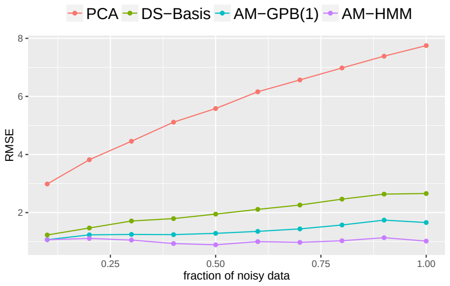

Fig. 5. RMSE in recovering true data using various decomposition methods as the fraction of noisy outliers in the data is increased.

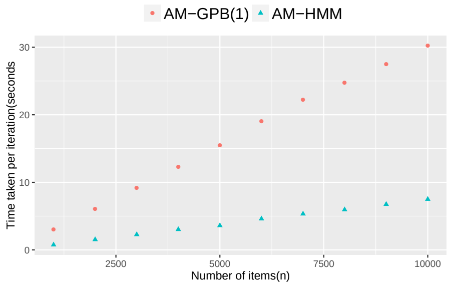

Fig. 6. Time taken per iteration for the two methods of computing smooth *DS*-basis

random. We want to see now if one can recover true data: WH. We will look the Root Mean Square Error(rmse); because of the construction of M, an algorithm that can recover the true H can achieve an rmse of 1 from WH on average, because of ϵ. But that requires being able to work through missing data and outlier noise μ.

Figure 5 shows the rmse achieved by different methods as the fraction of outliers f is varied. We see that AM-HMM is remarkably robust to noise and can recover the true basis even with many outliers. AM-GPB(1) is also close but as f is increased, it does slightly worse in recovering the basis. PCA does not work well at all in this scenario, and computing *DS*-basis without temporal regularization performs much worse as f increases. This illustrates why in real-world data with many outliers having temporal regularization is crucial when we know the underlying basis is smooth.

Figure 6 compares the execution time of AM-HMM and AM-GPB(1) as the rows of M are varied from 1K to 10K. Even though asymptotically, the two approaches have linear running time, in our experience AM-HMM is the only one that scales well for large groups, and we can see this in the rapidly increasing difference as we approach 10K items in the plot.

References

[1] S. Makridakis, S. C. Wheelwright, and R. J. Hyndman, *Forecasting methods and applications*. John Wiley & Sons, 2008.

[2] W. A. Fuller, *Introduction to statistical time series*, vol. 428. John Wiley & Sons, 2009.

[3] P. J. Brockwell and R. A. Davis, *Time series: theory and methods*. Springer Science & Business Media, 2013.

[4] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, 2009.

[5] Y. Bar-Shalom and X.-R. Li, "Estimation and tracking- principles, techniques, and software," *Norwood, MA: Artech House, Inc, 1993.*, 1993.

[6] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "Stl: A seasonal-trend decomposition procedure based on loess," *Journal of Official Statistics*, vol. 6, no. 1, pp. 3–73, 1990.

[7] A. Gelman and J. Hill, *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press, 2006.

[8] A. Jha, S. Ray, B. Seaman, and I. S. Dhillon, "Clustering to forecast sparse time-series data," in *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*, pp. 1388–1399, IEEE, 2015.

[9] W. Sun and D. Malioutov, "Time series forecasting with shared seasonality patterns using non-negative matrix factorization," in *NIPS Time Series Workshop*, 2015.

[10] J. W. Taylor, L. M. De Menezes, and P. E. McSharry, "A comparison of univariate methods for forecasting electricity demand up to a day ahead," *International Journal of Forecasting*, vol. 22, no. 1, pp. 1–16, 2006.

[11] M. Asteris, D. Papailiopoulos, A. Kyrillidis, and A. G. Dimakis, "Sparse pca via bipartite matchings," in *Advances in Neural Information Processing Systems*, pp. 766–774, 2015.

[12] V. Goyal and R. Ravi, "An fptas for minimizing a class of low-rank quasi-concave functions over a convex set," *Operations Research Letters*, vol. 41, no. 2, pp. 191–196, 2013.

[13] M. Asteris, D. S. Papailiopoulos, and G. N. Karystinos, "The sparse principal component of a constant-rank matrix," *IEEE Transactions on Information Theory*, vol. 60, no. 4, pp. 2281–2290, 2014.

[14] G. N. Karystinos, "Optimal algorithms for binary, sparse, and l 1-norm principal component analysis," in *Mathematics Without Boundaries*, pp. 339–382, Springer, 2014.

[15] Z. Ghahramani and G. E. Hinton, "Variational learning for switching state-space models," *Neural Computation*, vol. 12, no. 4, pp. 831–864, 2000.

[16] C.-J. Kim, "Dynamic linear models with markov-switching," *Journal of Econometrics*, vol. 60, no. 1-2, pp. 1–22, 1994.

[17] K. P. Murphy, "Switching kalman filters," tech. rep., Citeseer, 1998.