A Clustering Based Forecast Engine for Retail Sales

Vijayalakshmi Murlidhar[1], Bernard Menezes[1], Mihir Sathe[2], Goutam Murlidhar[2]
[1]Indian Institute of Technology (IIT)
Mumbai
India
[2]VES institute of Technology
India
{viji.murli, mihsathe}@gmail.com, bernard@it.iitb.ac.in

Journal of Digital Information Management

ABSTRACT: *Efficient and accurate sales forecasting is a vital part of creating an efficient supply chain in enterprises. Times series methods are a popular choice for forecasting demand sales. A major challenge is to develop a relatively inexpensive and automated forecasting engine that guarantees a desired forecasting accuracy. Times series decomposition and Forecast combination have been two classes of methods that have attracted the interest of recent researchers. One solution has been to use decomposition followed by recombining to form a very large number of forecasting models Further many recent papers present Data Mining based methods to intelligently discover a subset of methods from a large list that can be used in combination for sales demand forecasting. These methods are computationally expensive and prohibitive if they are applied to each individual time series in a retail organization. In this paper we present a novel technique to identify similar sales series and efficiently use the best combination of methods learnt for one series to forecast for the entire set of similar series.*

Categories and Subject Descriptors: I.5 [**Pattern Recognition**]; Pattern analysis: **I.5.3** [**Clustering**] **J.1** [**Administrative Data Processing**]; Marketing

General Terms: Forecasting, Pattern Mining, Data clustering

Keywords: Sales Forecasting, Times Series, Decomposition, Combining, Frequent Pattern Mining, Hierarchical Clustering, Good Experts, Bad Experts

Received: 9 May 2012, Revised 2 June 2012, Accepted 14 June 2012

1. Introduction

Accurate sales forecasting is of paramount importance to intelligent Supply Chain Management. Organizations investing in forecasting best practices gain a competitive advantage over rival organizations. For demand sales, times series based forecasting methods have been popular and well researched [1] [2]. The most widely used statistical models have been the exponential family of models and the ARIMA class of models [1] [2]. The biggest challenge in forecasting a sales series, is to identify the best forecasting model for a given times series. Sales series are normally too complex for one model to work best at all times. A realistic situation that could occur is, one could have a large set of non-optimal models for forecasting and choosing the best one is not really an option. Thus researchers have long since been advocating the use of a '*Combination Forecast*' [3], [4] [5]. In combination forecasting, instead of trying to choose the single best method, there is an attempt to identify a group of methods which would in conjunction, help to improve forecast accuracy [5]. Another approach that has met with much success is to first decompose a time series into its "*natural components*" and then do the forecasting.

Recent work in this area has effectively used series decomposition as a pre-processing technique, to forecast the individual components, followed by recombining of these individual forecasts to form the final forecast [6]. In the case of sales data, the natural components include trend, seasonality, and an irregular component. Each component is considered as an individual series and a battery of models is utilized to forecast each component series. Each decomposed series is forecasted using a set of models. These are recombined to generate a large number of combination forecast values per point. Computing a point forecast based on such a large number of estimates poses both an opportunity and a challenge. Data Mining provides us with a solution to this problem. Over the last fifteen years or so, a rich store of techniques has been compiled to mine large data repositories for valuable patterns, associations, correlations, trends, etc. Recent work by the same authors (paper submitted), uses frequent pattern mining to "*learn*" a group of

forecasters that tend to have superior forecasting performance for a part of the series (during the training phase). These forecasters are further used, to forecast the rest of the series [17]. One method identifies a set of *"good"* Seasonal, Trend and Irregular experts for a given series [19]. The complimentary problem of identifying a poorly-performing set of forecasters and then eliminating these during the testing phase has also been tested [19]. Both these algorithms have led to increased accuracy over the traditional standard model for demand sales, the Holt–Winter method [2].

Huge retail chains have thousands of items in their inventory. The task of forecasting the sales of each and every product using the frequent pattern mining based algorithm is computationally expensive and practically infeasible. In this paper we propose an extension to our frequent pattern mining based work, to overcome this problem. We use a hierarchical clustering algorithm to identify clusters within the inventory consisting of items which have the similar sales patterns. An innovative distance measure is proposed to measure the dissimilarity (distance) between two sales time series. Using our methods based on frequent pattern mining, we identify the *"good"* and *"bad"* Trend, Seasonal and Irregular models for one representative item of each cluster. These set of models are used to forecast sales for all items of the cluster. We propose to show that our algorithm in addition to being efficient also reduces the forecasting error considerably in comparison to the standard Holt Winter approach.

2. Existing Work

The selection and implementation of a proper forecast methodology for customer demand is always an important issue for most enterprises, since the productivity of the entire enterprise can rely on the accuracy of the forecast. A significant forecast error either below or above the actual value, may result in the firm landing up with excess inventory carrying costs or else lost customers due to item shortages. When demand is fairly stable, e.g., unchanging or else growing or declining at a known constant rate, choosing an appropriate forecast method and making an accurate forecast is less difficult. If, on the other hand, the sales pattern is erratic, the complexity of the forecasting task is compounded. Most sales series exhibit the latter characteristic [7], [8].

Employing multiple forecasting models captures the various irregularities that exist in a sales series. Two approaches to handle multiple forecasting models are model selection and model combination. The former involves choosing a single best model at each point, typically based on performance in the training phase. In model combination a subset of models are chosen and their average gives the forecast at that point. Combining, as an alternative to model selection, has been advocated since long [3] [4] [5] A detailed analysis of the results of the M1, M2 and M3 competitions, (premier competitions in the forecasting domain) by forecasting experts revealed that: *"simple methods seem to do at least as well as more complex methods, and combining forecasts helps, particularly for longer horizons"* [9], [10] [11] and [12]. Typically, demand sales series have several inherent patterns and it may not be possible to find, the single best method for forecasting [7], [8]. The most compelling motivation for using a combination to forecast comes from the fact that, enterprises would like to automate and speedup the forecasting process. A combination forecast can use vast number of simple sub-optimal models, and what is needed is an intelligent way to select and combine a subset of these models for a series. Combining a set of simple and similar models decreases the risk of forecasting and increases accuracy.

The use of series decomposition in conjunction with combining was pioneered in [6], [7] and further developed by the authors in [16], [17]. This work is characterized by the fact that a very large number of forecasters are used and a subset of these chosen for the final forecast. Using this large number of forecasters as a base, data mining based methods are used to learn a set of good and bad models for a particular series. Using this information considerably increased the forecast accuracy. This work is discussed in the next section.

2.1 Basic Forecasting Models

A time series is a sequence of observations that are collected, observed or recorded at successive intervals of time. An intrinsic feature of a time series is that, typically a set of adjacent observations are dependent. In times series based forecasting, the prediction is based on an inferred study of past data behaviour over time, i.e., the extrapolation method. An important requirement for such methods is to use reliable data. The availability of extensive retail scanner data from enterprises means that reliable data can be obtained for existing products. Simple classical approaches to times series forecasting include the methods based on Averaging and Smoothing. Two of the best known methods of forecasting seasonal data (such as retail sales) are the Holt-Winter method and seasonal ARIMA [1] [2].

The Holt-Winter method uses exponential smoothing of *level* (S_t), *trend* (T_t) and *seasonal index* (I_t) for forecasting the given series. For multiplicative seasonality, the model assumes the form

$$S_t = \alpha\left(\frac{X_t}{I_{t-c}}\right) + (1-\alpha)(S_{t-1} + T_{t-1})$$

$$T_t = \beta(S_t - S_{t-1}) + (1-\beta)T_{t-1}$$

$$I_t = \gamma\left(\frac{X_t}{S_t}\right) + (1-\gamma)I_{t-c}$$

Here, c is the number of observation points in a cycle ($c = 12$ for monthly data). α, β and γ are the smoothing constants with values between 0 and 1. The forecast at time t for the series at time $(t + i)$ is $(S_t + iT_t)\, I_{t-c+i}$.

The general multiplicative seasonal ARIMA $(p, d, q) \times (P, D, Q)$ model is expressible as

$$\Phi(B)\varnothing(B^c)(1-B^d)(1-B^c)^D X(t) = C_0 + \theta(B)\Theta(B^c)\varepsilon(t), \quad t = 1, 2,...$$

Here C_0 is a constant, $\varepsilon(t)$ is a sequence of independent, zero mean and normally distributed errors, d and D are the orders of non-seasonal and seasonal differencing for the time series and $\Phi(B)$, $\varnothing(B^c)$, $\theta(B)$ and $\Theta(B^c)$ operators are polynomials in B (the backward shift operator) with the following general forms

$$\Phi(B) = 1 - \Phi_1(B) - \Phi_2(B^2)... - \Phi_p(B^p)$$
$$\varnothing(B^c) = 1 - \varnothing_1(B^c) - \varnothing_2(B^{2c})... - \varnothing_p(B^{pc})$$
$$\theta(B) = 1 - \theta_1(B) - \theta_2(B^2)... - \theta_p(B^q)$$
$$\Theta(B^c) = 1 - \Theta_1(B^c) - \Theta_2(B^{2c})... - \Theta_q(B^{qc})$$

The polynomials $\Phi(B)$ and $\theta(B)$ capture the non-seasonal behaviour and $\varnothing(B^c)$ and $\Theta(B^c)$ capture the seasonal behaviour of the series. The differencing orders d and D typically have a value 0 or 1. For non-seasonal ARIMA, $P = D = Q = 0$.

There is a need to evaluate the performance of a forecasting model in terms of the error it introduces. For demand sales, the most popular way to evaluate a forecasting model is to use the Mean Absolute Percentage Error (MAPE) value [3]. The MAPE is defined as:

$$(100/n) \times \sum_{i=1}^{n} \frac{(|X'(t) - X(t)|)}{X(t)}$$

The Holt Winter (HW) Multiplicative approach is normally regarded as a standard approach for sales forecasting. All our results are compared with the accuracy (MAPE) of the HW method.

2.2 Rank Based Combining Methods

In recent times a new class of combination techniques, Rank Based Combining, have been proposed and researched in [6], [13] [16] and [17]. There has been sufficient empirical evidence to suggest that these new methods can increase accuracy of forecasts. These methods use basic ARIMA and smoothing models that are available in any statistical software.

These methods combined a decomposition followed by a combining step:

Each series is decomposed into its Trend, Seasonality and Irregular components using the expressions below. Let X_t, T_t, S_t and I_t denote the t^{th} point (i.e., t^{th} month) of the respective series. We define the Trend, T_t, at a point t as the average sales in 12 consecutive months up to and including month = t.

$$T_t = \frac{\sum_{i=0}^{11} X_{t-i}}{12}$$

The seasonal component is

$$S_t = \frac{\frac{X_t}{T_t} + \sum_{i=0}^{n-1} S_{t-12i}}{n} \text{ Where } n = \left\lceil \frac{t}{12} \right\rceil$$

The irregular component is simply

$$I_t = \frac{X_t}{T_t \times S_t}$$

A method used in forecasting a single component is referred to as an atomic forecaster. A forecaster for the original series is a triplet made up of the atomic forecasters for each component. The set of such triplets is the Cartesian product of the sets of forecasters for the T, S and I components. Each such triplet of atomic forecasters (T, S, I) is called an *"Expert"*. In this work, we use a total of 86 Trend models, 33 Seasonal models and 34 Irregular models. These are mostly ARIMA and seasonal ARIMA models of different orders and various Exponential Smoothing Models. The Cartesian product of the Trend, Seasonal and Irregular models gives rise to 96,492 experts. An expert forecasts each point in the series and so we have 96,492 forecasts per point. The Appendix includes a list of atomic forecasters used in this work.

There is a need to select a subset of good models from this large set of models to forecast the next point accurately. Rank based methods sort the different forecasters (96492) based on the MAPE value at that point. Some top K, of these forecasters is chosen to forecast the next point. This is repeated at every point. This method is called *"TopK"*. The important issue is how to choose the value K. In a variant of this method, *"DTopK"* (Dynamic Top K) [10], all 'K' values are evaluated and the best value of K that gives minimum MAPE value is chosen. These methods show a decrease in MAPE value when compared with the HW forecasts for the same series.

2.3 Use of Frequent Pattern Mining (FPM)

The rank based methods are cumbersome to use and very time consuming. The methods are trivial and do not effectively use the knowledge that exists in the times series and they fail to identify any pattern that could be exploited by a combination forecast.

As an extension to the rank based method a frequent pattern mining based algorithm was used to learn a set of good and bad experts for a series by the authors in [16] and [17]. Association (Frequent Pattern) mining is a standard way to identify patterns from massive datasets [14], [15]. This method uses a portion of the times series as training, to identify expert sets that are consistently good or bad. To forecast further points in the series, either the good set of experts are used or alternately, bad experts are filtered out and the surviving good set of experts are used. A brief overview of this work is presented in the next section.

2.4 Fine Grained Frequent Pattern Mining

A method proposed by the authors in [20], using Frequent Pattern Mining (FPM), termed Fine Grained FPM, (FG-FPM) generates a set of *"good"* Trend T, Seasonality S

and Irregular I experts for a series. In a similar fashion it is also possible to extract the *"bad"* T, S and I experts. These bad experts are filtered out of from the set of experts and the surviving experts used for forecasting. This approach is the Fine Grained Filtered Experts approach (FG-FE). The above two approaches are able to learn the good and bad experts for a series using just 50% of the initial points. The forecast accuracy by using the surviving set of T, S and I experts in combination improves over the Holt Winter forecast accuracy.

• FG- FPM

The initial "n" points of the series are used as the training set, "n" is the training size.

1. At point $(i) / \{1 <= i <= n\}$ take top 20000 experts based on MAPEs; Split each such good expert into its constituent S, T & I experts.

2. At each point "i" we construct 3 sets $T(i)$, $S(i)$, $I(i)$ that consist the expert numbers (identities) of the atomic forecasters for the trend, seasonality and irregular components, that have appeared in the top 20,000 experts at that point.

3. Since each T, S, I expert may occur several times in the top 20,000; we fix a threshold frequency and only consider those experts that cross this frequency.

4. We now have 3 sets:

$$T = \{ T(i) / \{1 <= i <= n \}$$
$$S = \{ S(i) / \{1 <= i <= n \}$$
$$I = \{ I(i) / \{1 <= i <= n \}$$

5. We use any standard frequent pattern mining [17], on the above 3 sets, with support value of 70% and confidence at 90% to generate the set of experts that have consistently occurred in the top 20,000 experts list. We get 3 sets of consistent good experts Tg, Sg and Ig.

6. We compute average of the experts in Tg to form $Tg\_average$. We similarly form $Sg\_average$ and $Ig\_average$.

7. The final forecast is the product of $Tg\_average$, and $Sg\_average$ and $Ig\_average$.

8. We use the same set of experts Tg, Sg and Ig for the rest of the series.

• FG- FE

In a similar fashion starting with the bottom 20,000 experts at each point, it is possible to use "n" points as training set and identify the set of bad atomic experts for the series, T_b, S_b and I_b. Filter (remove) these bad experts out and use the average of the surviving T S and I experts to form the final forecast for the rest of the series.

With a training size of n = 50% of the points, these algorithms were able to identify the set of good and bad experts for a series. Using these experts for forecasting the further points in the series resulted in a considerable improvement over the HW forecasting accuracy.

3. Clustering Based Forecast Engine

We now propose our improved design for an intelligent forecast engine that can be used in a retail business to forecast product demand sales. Retail enterprises have thousands of items on their inventory. It is imperative for enterprises to be to generate quick, timely and accurate forecasts for all these items. One way of generating accurate forecasts with minimum error would be to use our methods described in section 2. Using these methods for every item on the inventory list would be prohibitive with respect to the time taken to identify a surviving set of good forecasting models. We therefore propose an optimal solution to this problem by decreasing the number of times the FPM based algorithm must be used.

Our design works in two stages. In stage one we form clusters of all items in the inventory by grouping together all items that have similar sales patterns. We use the standard hierarchical agglomerative clustering algorithm to cluster the items. We introduce a new distance measure, based on sales pattern similarity to perform the clustering.

In stage two, we pick one item from each cluster and use its sales series to learn the best set of forecasting models based on methods in section 2. We use the same set of forecasting models for all items in that cluster. We illustrate the working and results of our algorithm using some real life datasets.

3.1 Stage 1: Clustering of Items

In this stage our objective is to group together, series (items) that follow similar sales patterns. Times series clustering has been a hotbed of research for some time now. Most of the work extends traditional methods of clustering, to time series also [21]. We use an agglomerative hierarchical approach to group items into clusters. A hierarchical clustering method works by grouping data objects into a tree of clusters. Agglomerative hierarchical methods start by placing each object in its own cluster and then merge clusters into larger and larger clusters, until all objects are in a single cluster, Figure 1. A termination condition can then be defined to identify clusters of interest.

In order to decide which clusters should be combined at each step, a measure of dissimilarity between sets of observations is required. In most methods of hierarchical clustering, this is achieved by use of an appropriate metric (a measure of distance between pairs of observations), which specifies the dissimilarity of sets as a function of the pairwise distances of observations in the sets. The usual distance measures like Euclidean or Manhattan distance will not be appropriate since we need to differentiate between two time series in terms of their sales patterns. One obvious choice would be to use correlation to find similarity in two series. But a high correlation coefficient between two series would not assure us of similarity in increasing or decreasing patterns. [18]. Yule [19] recognized that Correlation does not imply causation. If two variables X and Y have a high index of correlation, it does not necessarily mean that an increase in X will cause an increase in Y [19].

Thus we introduce a new measure of distance between two sales times series, where we take into account the rise and fall of sales to match two series. Visually we can consider two series to be similar if their graphs follow fairly similar increasing and decreasing patterns. Though it is possible to find similarities in patterns by observing the graphs, the main challenge is to automate this process and define a standard way to find the distance between two series. We call this new distance measure as Sales pattern Distance (SPD) between two series.

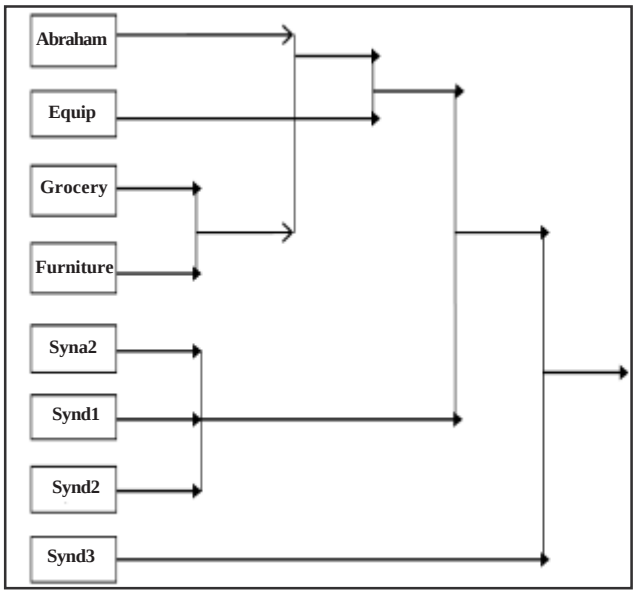

Figure 1. Agglomerative Hierarchical Clustering

3.1.1 Sales Pattern Distance (SPD)

We now define the steps required to calculate the SPD between two sales time series. Retail organization may have items whose units of sales differ very vastly in scale. For example, unit sales for television sets may be in hundreds per month as compared to audio CDs that may be sold in thousands or groceries that may be sold in hundreds of thousands per month. Rather than looking just at the volumes of increase or decrease per month we use only the percentage difference as a measure. We only use the quantity in percentage by which the successive values in the time series vary. Since percentages are independent of the scales of original series, they are comparable.

1. Initialize $SPD\,(X, Y) = 0$;

2. Start $i = 1$; Start from the same point for both series X_i and Y_i

3. For next point, $i+1$, we compute the movement from i, $(|X_i - X_{i+1}|, |Y_i - Y_{i+1}|)$. This is recorded as *"**up**"*, *"**down**"*, or *"**flat**"*, based on whether the sales figure went up, down, or remained the same. For very small changes say less than 5%, we assume the movement is ***flat***.

4. We also record the percentage of change from i point to point $i+1$ for both the series, X and Y

5. If the movement for both the series, from i point to point $i+1$ are not the same, we increment $SPD\,(X, Y)$ by 1.

6. If movement is same for both the series, that is both X and Y record the same movement (*up, down, flat*), we check the percentage of change from the previous point. If the difference in % change between X and Y is within a permissible threshold, we consider that both the series have shown a similar pattern at these two consecutive points. If the change is beyond the permissible threshold we increment the $SPD\,(X, Y)$ by 1.

7. Repeat steps 3 to 6, for the next set of points up to point *'n'* the desired training size.

i.e $\boldsymbol{SPD\,(X, Y)}$ is incremented at a point $'i+1'$, if

$$\boldsymbol{movemnet\,(|Xi - Xi + 1|)\ not = movement\,(|Yi - Yi + 1|)}$$

or if they are equal then

$$\boldsymbol{Difference\ in\ \%\ change\ ((|Xi - Xi + 1|), (|Yi - Yi + 1|)) > threshold.}$$

3.1.2 Hierarchical Agglomerative Clustering

The new distance measure proposed can be used to perform hierarchical clustering for all the items in the inventory. Let us assume that we have monthly sales figures (time series) for about *'m'* items in the inventory. These are denoted as $I1, I2, ..., Im$. Our aim is to cluster these series based on the SPD distance measure. Initially each series will be in its own cluster. The SPD as we have defined it earlier actually gives us a count of the number of points at which the two times series differ. We fix the threshold for the $SPD\,(X, Y)$ at each step as some 'τ' % of the total number of points where the two series have been compared.

At every step, this threshold 'τ' will be increased by a small value $(\tau + \Delta)$. This results in some of the clusters at the previous step being merged into larger clusters. This is continued until at some stage all the series merge into one cluster. By evaluation of the clusters or using some domain knowledge we could decide to stop clustering either when the clusters are too far apart to be merged (distance criterion) or when there is a sufficiently small number of clusters (number criterion). In this work we choose to stop clustering when the threshold reaches a particular level of dissimilarity after which our forecasting results in poor results.

3.2 Stage 2

We now have clusters of items based on similar sales patterns. Let us assume that we stop clustering at *'k'* clusters. We pick one Candidate Series from each cluster *'i'*, $CS\,(i)$.

Using the Fine Grained algorithms described in section 2.4, we learn a set of *"good"* and *"bad"* Experts for our series.

1. We first decompose the series $CS\,(i)$ from each cluster *'i'*, into its corresponding Trend, Seasonality and Irregular (T, S, I) components.

2. For each $CS\,(i)$ apply 86 Trend models, 33 Seasonal models and 34 Irregular models. The description of the different models is given in the appendix.

3. Taking the Cartesian product of each of these forecasts we obtained 96492 forecasts for each point.

4. From this pool of forecasts we identify the top 20,000 experts (forecasters) based on the MAPE values, for each point. We fix a training size, *'m'* to decide how far into the series we need to consider, to learn the consistent good or bad experts of the series.

5. For every point *'x'* in the series, starting from 25th point to *mth* point, we form sets, *Tgood* [*x*], *Sgood* [*x*] and *Igood* [*x*] consisting of the *T, S* and *I* models that appeared in the top 20,000 forecasts at point *x*.

6. We use a standard frequent pattern mining algorithm, with appropriate values for support and confidence, on the sets:

i. *Tgood* [*x*] = {25 < = *x* < = *m* }

ii. *Sgood* [*x*] = {25 < = *x* < = *m* }

iii. *Igood* [*x*] = {25 < = *x* < = *m* }

7. We obtain a set of consistently good *T, S* and *I* experts for each series *CS* (*i*)

8. We have thus identified a set of cluster specific good experts for all series of the cluster *'i'*.

9. For all series in a cluster *'i'*, apply the same set of good *T, S* and *I* experts to forecast the next point. .

10. In a similar fashion we identify the set of bad experts for a cluster by identifying the bad set of experts for one series in the cluster. We filter these bad experts and using the surviving set of experts for all series in the cluster.

11. To evaluate the accuracy of our algorithm, we recorded the MAPEs obtained for the last 24 points of each series, and compared it with the MAPE of the Holt winter forecast of the last 24 points of the series.

4. Experimental Results

In this section we discuss the results of applying our algorithm on real life datasets. Empirical results indicate that for all series in the dataset, our method improves forecast results. We use the Holt Winter method as benchmark to which to compare the forecasting performance since, this method is he standard method used for demand sales.

4.1 Dataset

Our series are picked form real life dataset US retail sales from Economagic.com [23]. All series are monthly sales figures of different product from US retail data. All series have data starting from January 1980 to December 2010. We picked 25 series from this set to test the results of our hypothesis. The list of series is shown in figure 3.

4.2 Clustering of Series

The hierarchical agglomerative algorithm is used on these 25 series. Table 1 illustrates a sample run of the computation of the *SPD* between two sample series from our dataset, Drugs and Health. We look at both the direction of movement and also the percentage difference with respect to the previous point to identify whether two series have *"same"* movements or not. When the movements for the series are not the same we increment the SPD value.

The visual plot of the two series indicates that these two series have similar rising and falling patterns, Figures 2A Figure 2B.

The final results for the two series were as follows:

Total points: 235
No. of points where the 2 series moved in same direction and % change within threshold: 225
No. of points where they were in opposition: 8
Thus SPD (Drugs, Health) = 8
i.e. 4% difference and hence they ultimately belong to the same cluster.

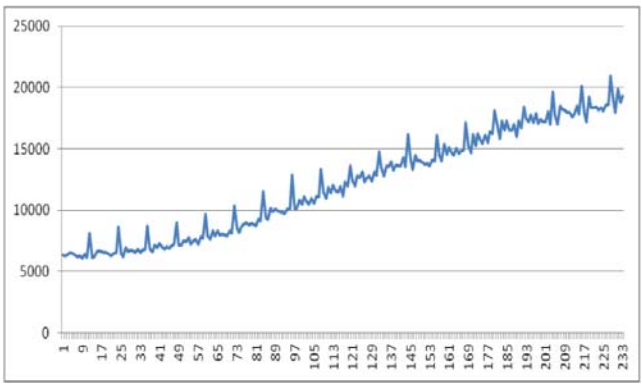

Figure 2A. Drugs Monthly Sales Series

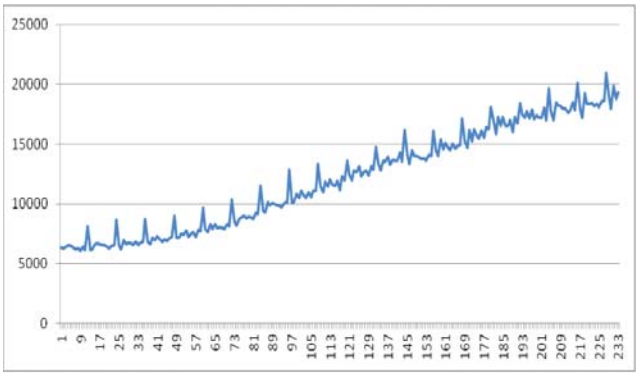

Figure 2B. Health Monthly Sales Series

Initially all the 25 series are in its own cluster, the threshold is 0%, i.e. *SPD* = 0. We then increase the threshold step by step 10% at a time. We continue upto all series merge into one cluster. At each step we compute a distance matrix between existing clusters using the *SPD* as a distance measure and single linkage. Some stages of this clustering are indicated in the figures 3 and 4.

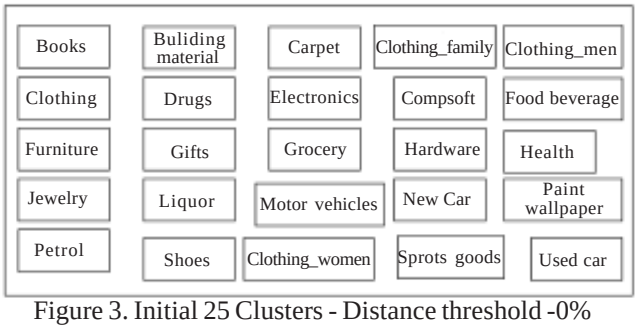

Figure 3. Initial 25 Clusters - Distance threshold -0%

Point No.	Series Drugs	% Change	Series Health	% Change	Difference	Movement.
22	6881	7554	-----			
23	6356	0.84	7532	-0.89	1.13	Diff
----	----	----	----	----	----	----
----	----	----	----	----	----	----
70	8141	-1.99	9718	-3.15	1.16	Same
71	10378	27.48	12232	25.87	1.61	Same
77	8787	-2.03	10663	-0.70	1.33	Same
78	8899	1.27	10774	1.04	0.23	Same
79	8887	-0.13	10766	-0.07	0.06	Same
80	8733	-1.73	10469	-2.76	1.03	Same
81	9235	5.75	11006	5.13	0.62	Same
82	9148	-0.94	10817	-1.72	0.78	Same

Table 1. Sample Run of SPD Computation

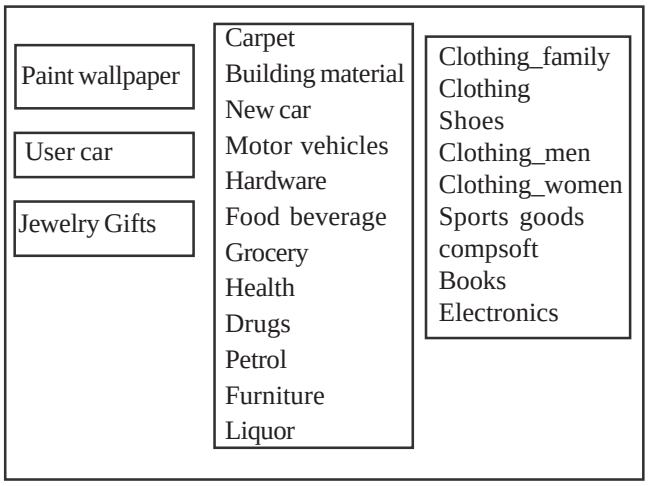

Figure 4. Five Clusters - Distance threshold -25%

The distance matrix at the stage where 7 clusters merge into 5 clusters is depicted in Table 2. When we tested the results of our forecasting at every stage of clustering, the accuracy of forecasting is maximum when the threshold distance is about 25% points i.e. the two series are dissimilar at 25% of the points or similar 75% of the time. We stop clustering at this stage.

4.3 Learning the forecasters for each Cluster

Now that we have the five final clusters of series, we pick any one representative series from each cluster, *CS*(1), *CS*(2), *CS*(3), *CS*(4), *CS*(5).

Let us consider *CS*(4) as the series *'Drugs'*; we now try and learn the set of consistent good *T, S* and *I* experts for this series.

1. We start from 25th point of series Drugs and go up to 50% of the series, point 100.

2. For each point *'i'*, we identify the top 20,000 experts (*T, S, I*). We sort all these experts based on the Absolute Percentage error (*APE*) values of their forecasts at that point.

$$APE(i) = Abs\frac{(Actual(i) - Forecast(i))}{Actual(i)}$$

3. We collect these $T(good)(i)$, $S(good)(i)$ and $I(good)(i)$ at all points up to $i = 100$;

4. We perform frequent pattern mining on these sets, with support of 70% and confidence of 90%. We obtain the set consistent of good *T, S* and *I* experts for the Drugs series.

Cluster	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7
Cluster 1	00	46	36	43	45	52	60
Cluster 2	46	00	33	39	50	46	41
Cluster 3	36	33	00	37	46	45	52
Cluster 4	43	39	37	00	39	45	53
Cluster 5	45	50	46	39	00	62	52
Cluster 6	52	46	45	45	62	00	47
Cluster 7	60	41	52	53	52	47	00

Table 2. SPD values at 7 Clusters

5. We have used the CHARM [20] algorithm from open source package SPMF [22], to implement this algorithm.

For the series Drug some of the experts that were found to be consistent were:

$$T(good) = \{T2, T3, T6, T11, T51, T8, \ldots \ldots . T53, T5, T71\}$$

$$S(good) = \{S2, S4, \ldots \ldots \ldots S8, S17 \ldots S33\}$$

$$I(good) = \{I1, I4, I5, I8, I10, I12, \ldots \ldots \ldots\}$$

In a similar fashion we find the best set of *T, S* and *I* experts for the series Paint, Usedcar, Jewelry, and Clothing for the clusters 1, 2, 3, and 5 respectively. We also find the set of bad experts for these series using the bottom 20,000 experts at each point.

4.4 Forecasting For All Series

For each series in our list, we now compute the forecasts from the 25th point onwards using the good T, S and I experts identified for the cluster they belong to. These are the experts identified for the representative series for each cluster.

1. Cluster 1: **Paint** wallpaper
2. Cluster 2: **Used car**
3. Cluster 3: **Jewelry**, Gifts
4. Cluster 4: **Drugs**, Carpet, Building material, New car, Motor vehicles, Hardware, Food beverage, Grocery, Health, Petrol, Furniture
5. Cluster 5: **Clothing**, Clothing_family, Shoes, Clothing_men, Clothing_women, Sports goods, Compsoft, Books, Electronics

$$Forecast(i_{good}) = (T_{avg}(i) * (I_{avg}(i) * (S_{avg}(i)) \; where$$

{$T_{avg}(i)$ = Average of all the Trend values obtained at point '*i*', by Good T experts

$S_{avg}(i)$ = Average of all Seasonality values obtained at point '*i*', by Good S experts

$I_{avg}(i)$ = Average of all the Irreg. values obtained at point '*i*', by Good I experts}

Series	HW	Forecast (good)	% Improv-ement	Forecast (bad)	% Improv-ement
Books	4.68	3.98	14.96	3.96	15.38
Building	3.01	2.56	14.95	2.34	22.26
Carpet	3.34	2.98	10.78	3.01	9.88
Clothing	2.42	2.23	7.85	2.13	11.98
clothing_family	2.83	2.89	-2.12	2.82	0.35
clothing_men	3.21	3.34	-4.05	3.23	-0.62
clothing_women	2.85	1.7	40.35	1.9	33.33
compsoft	3.67	3.77	-2.72	3.67	0.00
Drugs	1.55	1.24	20.00	1.34	13.55
electronics	2.05	2.00	2.44	2.00	2.44
foodbeverage	1.21	0.99	18.18	0.97	19.83
furniture	2.22	2.12	4.50	2.13	4.05
Gifts	4.71	3.98	15.50	3.67	22.08
hardware	2.37	2.05	13.50	2.03	14.35
Health	1.45	1.29	11.03	1.22	15.86
jewelry	3.77	3.44	8.75	3.23	14.32
Liquor	2.19	2.01	8.22	1.98	9.59
motorvehicles	3.43	3.45	-0.58	3.43	0.00
newcar	3.85	3.56	7.53	3.58	7.01
petrol	3.17	3.04	4.10	3.07	3.15
Shoes	2.93	2.88	1.71	2.94	-0.34
sportgoods	2.35	2.23	5.11	2.21	5.96
usedcar	3.56	3.45	3.09	3.34	6.18
grocery	1.20	1.19	0.83	1.1	8.33
paintwallpaper	3.81	3.23	15.22	3.14	17.59
Improvement%			**8.77%**		**10.26%**

Table 3. Comparing Accuracy with Holt Winter

$$Forecast\,(i_{bad}) = (T_{avgb}\,(i) * (I_{avgb}\,(i) * (S_{avgb}\,(i)) \; where$$

{$T_{avgb}\,(i)$ = Average of all the Trend values obtained at point 'i', by filtering the bad T experts

$S_{avgb}\,(i)$ = Average of all Seasonality values obtained at point 'i', by filtering the bad S experts

$I_{avgb}\,(i)$ = Average of all the Irreg. values obtained at point 'i', by filtering the bad I experts}

We compute the MAPE values for the last 24 points (two years) for all these series for both these forecasts. We compare this with the MAPE value obtained for the same set of points using the Holt Winter method. Table 3 presents these results. We can see from the table that our method shows an improvement in forecasting accuracy for most of the series over that of the single method Holt Winter.

5. Conclusions

In this paper we have presented an innovative and practical way of generating accurate forecasts for demand sales that could be useful in a retail environment. We consider a large number of fairly performing models and identify a set of consistent good and bad models from them. Each model is a standard model that is normally a part of any statistical software package. The method is highly efficient and effectively uses simple data mining techniques to learn the consistent set of experts. In a retail scenario with thousands of items in the inventory, it is expected that several items will have similar sales patterns. Our distance measure captures the similarities and dissimilarities in the sales pattern quite accurately.

By generating a good set of forecasters for an entire cluster by looking at just one series in the cluster is a huge optimization. Our algorithm will reduce the time to forecast effectively and help in generating just in time forecasts. In a typical retail enterprise, the size of each cluster could well be over a thousand. In such cases it would be an over simplification to assume one series could represent the entire cluster. One major extension would be to try and generate a set representative series for a cluster and use all of them to find cluster specific good and bad experts.

It would also be interesting to test our algorithm in the area of stock market prediction, where the series are very volatile and any pattern may be short lived. In such a scenario, we can try a windowing approach, which would periodically try to learn a clusters and also the consistent set of experts for a particular window period rather than the entire series.

6. Appendix : Details Of the Experts Used

ID	Expert Name	ID	Expert Name
1	ARIMA(0,0,1)(0,1,1)s	17	Log ARIMA(0,0,1)(0,1,1)s
2	ARIMA(0,0,2)(0,1,1)s	18	Log ARIMA (0,0,2)(0,1,1)s
3	ARIMA(0,1,1)(0,1,1)s	19	Log ARIMA (0,1,1)(0,1,1)s
4	ARIMA(0,1,1)s	20	Log ARIMA (0,1,1)s
5	ARIMA(0,1,2)(0,1,1)s	21	Log ARIMA (0,1,2)(0,1,1)s
6	ARIMA(1,0,0)(0,1,1)s	22	Log ARIMA (1,0,0)(0,1,1)s
7	ARIMA(1,0,1)(0,1,1)s	23	Log ARIMA(1,0,1)(0,1,1)s
8	ARIMA(1,1,0)(0,1,1)s	24	Log ARIMA(1,1,0)(0,1,1)s
9	ARIMA(1,1,1)(0,1,1)s	25	Log ARIMA(1,1,1)(0,1,1)s
10	ARIMA(1,1,2)(0,1,1)s	26	Log ARIMA(1,1,2)(0,1,1)s
11	ARIMA(2,0,0)(0,1,1)s	27	Log ARIMA(2,0,0)(0,1,1)s
12	ARIMA(2,1,0)(0,1,1)s	28	Log ARIMA(2,1,0)(0,1,1)s
13	ARIMA(2,1,1)(0,1,1)s	29	Log ARIMA(2,1,1)(0,1,1)s
14	ARIMA(2,1,2)(0,1,1)s	30	Log ARIMA(2,1,2)(0,1,1)s
15	ARIMA(3,0,0)(0,1,1)s	31	Log ARIMA(3,0,0)(0,1,1)s
16	ARIMA(3,1,0)(0,1,1)s	32	Log ARIMA(3,1,0)(0,1,1)s
		33	Holt Winter

Table A1. Seasonal Experts

ID	Expert Name	ID	Expert Name
1	ARIMA (0,0,1)s	18	Log ARIMA (0,0,1)s
2	ARIMA (0,1,0)	19	Log ARIMA (0,1,0)
3	ARIMA (0,1,1)	20	Log ARIMA(0,1,1)(1,0,0)s NOINT
4	ARIMA(0,1,1)(1,0,0)s NOINT	21	Log ARIMA (0,1,1)s NOINT
5	ARIMA (0,1,1)s NOINT	22	Log ARIMA (1,0,0)
6	ARIMA (1,0,0)	23	Log ARIMA (1,0,0)s
7	ARIMA (1,0,0)s	24	Log ARIMA (1,0,1)s
8	ARIMA (1,0,1)s	25	Log ARIMA (1,1,0)
9	ARIMA (1,1,0)	26	Log ARIMA (1,1,2)
10	ARIMA (1,1,2)	27	Log ARIMA (2,0,0)
11	ARIMA (2,0,0)	28	Log ARIMA(2,0,0)(1,0,0)s
12	ARIMA (2,0,0)(1,0,0)s	29	Log (3,1,1) NOINT
13	ARIMA (3,0,0)(1,0,0)s	30	Log Linear Exponential
14	Linear Exponential	31	Log Linear Trend AR1
15	Linear Trend AR1	32	Log Linear Trend AR2
16	Linear Trend AR2	33	Log Linear Trend AR3
17	Linear Trend AR3	34	Random

Table A2. Irregular Experts

Expert ID	Expert Name	Expert ID	Expert Name	Expert ID	Expert Name	Expert ID	Expert Name
1	ARIMA (0,1,0)(0,0,1)s	11	ARIMA (0,2,1) NOINT	21	ARIMA (1,1,1) NOINT	31	ARIMA (2,1,0)
2	ARIMA (0,1,0)(1,0,0)s	12	ARIMA (1,0,1)	22	ARIMA (1,1,2)	32	ARIMA (2,1,0)(1,0,0)s
3	ARIMA (0,1,0)(1,0,0)s NOINT	13	ARIMA (1,1,0)	23	ARIMA (1,1,2)(0,0,1)s	33	ARIMA (2,1,0)(1,0,0)s NOINT
4	ARIMA (0,1,0)(1,0,1)s	14	ARIMA (1,1,0)(0,0,1)s	24	ARIMA (1,1,2)(1,0,0)s	34	ARIMA (2,1,0) NOINT
5	ARIMA (0,1,1)	15	ARIMA (1,1,0)(1,0,0)s	25	ARIMA (1,1,2) NOINT	35	ARIMA (2,1,1)
6	ARIMA (0,1,1)(1,0,0)s NOINT	16	ARIMA (1,1,0)(1,0,0)s NOINT	26	ARIMA (1,2,0)	36	ARIMA (2,1,1) NOINT
7	ARIMA (0,1,1) NOINT	17	ARIMA (1,1,0)(1,0,1)s	27	ARIMA (1,2,0) NOINT	37	ARIMA (2,1,2)
8	ARIMA (0,1,2)	18	ARIMA (1,1,0) NOINT	28	ARIMA (1,2,1)	38	ARIMA (2,1,2) NOINT
9	ARIMA (0,1,2) NOINT	19	ARIMA (1,1,1)	29	ARIMA (1,2,1) NOINT	39	ARIMA (2,2,1)
10	ARIMA (0,2,1)	20	ARIMA (1,1,1)(0,0,1)s	30	ARIMA (2,0,1)	40	ARIMA (2,2,1) NOINT

Table A3. Trend Experts

Expert ID	Expert Name	Expert ID	Expert Name	Expert ID	Expert Name	Expert ID	Expert Name
41	ARIMA (3,1,0)	53	Log ARIMA (0,1,2)	65	Log ARIMA (1,1,1) NOINT	77	Log ARIMA (2,1,0) NOINT
42	ARIMA (3,1,0)(0,0,1)s	54	Log ARIMA (0,1,2) NOINT	66	Log ARIMA (1,1,2)	78	Log ARIMA (2,1,1)
43	ARIMA (3,1,0)(1,0,0)s	55	Log ARIMA (0,2,1)	67	Log ARIMA (1,1,2)(0,0,1)s	79	Log ARIMA (2,1,1) NOINT
44	ARIMA (3,1,0) NOINT	56	Log ARIMA (0,2,1) NOINT	68	Log ARIMA (1,1,2)(1,0,0)s	80	Log ARIMA (2,1,2)
45	Holt	57	Log ARIMA (1,1,0)	69	Log ARIMA (1,1,2) NOINT	81	Log ARIMA (2,1,2) NOINT
46	Log ARIMA (0,1,0)(0,0,1)s	58	Log ARIMA (1,1,0)(0,0,1)s	70	Log ARIMA (1,2,0)	82	Log ARIMA (2,2,1)
47	Log ARIMA (0,1,0)(1,0,0)s	59	Log ARIMA (1,1,0)(1,0,0)s	71	Log ARIMA (1,2,0) NOINT	83	Log ARIMA (2,2,1) NOINT
48	Log ARIMA (0,1,0)(1,0,0)s NOINT	60	Log ARIMA (1,1,0)(1,0,0)s NOINT	72	Log ARIMA (1,2,1)	84	Log ARIMA (3,1,0)
49	Log ARIMA (0,1,0)(1,0,1)s	61	Log ARIMA (1,1,0)(1,0,1)s	73	Log ARIMA (1,2,1) NOINT	85	Log ARIMA (3,1,0)(0,0,1)s
50	Log ARIMA (0,1,1)	62	Log ARIMA (1,1,0) NOINT	74	Log ARIMA (2,1,0)	86	Log ARIMA (3,1,0)(1,0,0)s
51	Log ARIMA (0,1,1)(1,0,0)s NOINT	63	Log ARIMA (1,1,1)	75	Log ARIMA (2,1,0)(1,0,0)s		
52	Log ARIMA (0,1,1) NOINT	64	Log ARIMA (1,1,1)(0,0,1)s	76	Log ARIMA (2,1,0)(1,0,0)s NOINT		

Table A4. Trend Experts Contd.

References

[1] Brockwell, P. J., Davis, R. A. (1991). *Time series: Theory and Method*, (Second edition. Springer International Edition).

[2] Makridakis, S., Wheelwright, S., Hyndman, R. (1998). *Forecasting methods and Applications*. (Third Edition. Wiley).

[3] Robert, Clemen, T. Combining forecasts: A Review and Annotated Bibliography. *International Journal of Forecasting.*

[4] Granger C., Ramanathan, R. (1984). Improved methods of combining forecasts. *Journal of Forecasting*, 3, 197–204.

[5] Scott Armstrong, J. (1990). Combining forecasts: The end of the beginning or the beginning of the end? *International Journal of Forecasting,* 5(4) 585–588, October. (Reformatted September 2006)

[6] Menezes, B., Seth, A., Singh, R. (2007). Can a million experts improve your sales' forecasts? *European Symposium on Time Series Prediction*.

[7] Carlo Altavilla, Matteo Ciccarelli, (2007). Information combination and forecast (st)ability - Evidence from vintages of time-series data, ECB Working Paper Series, December: (846), http://ssrn.com/abstract_id=1068862.

[8] Michael, J., Baker. (1999). Sales Forecasting, The *IEBM Encyclopaedia of Marketing*, International Thompson Business Press, p. 278-290.

[9] Jeremy Smith, Kenneth, F., Wallis (2005). Combining Point Forecasts: The Simple Average Rules, OK?, February, University of Warwick.

[10] Makridakis, S., Hibon, M. (2000). The M3-Competition: Results, Conclusions and Implications, *International Journal of Forecasting,* 16 (4) 451–476.

[11] Scott Armstrong, J. (2001). Combining Forecasts. *Principles of Forecasting: A Handbook for researchers and Practitioners,* Scott Armstrong, J. (ed.): Norwell, MA: Kluwer Academic Publishers.

[12] Timmermann, A. (2006). Forecast Combinations, *In:* eds. G. Elliott, C.W.J. Granger and A. Timmermann, *Handbook of Economic Forecasting*, Elsevier Press.

[13] VenuGopal, (2007). Forecasting using consistent experts, Masters Thesis, Kanwal Rekhi School of Information Technology, Bombay, INDIA.

[14] Agrawal, R., Imielinski, T., Swami, A. (1993). Mining association rules between sets of items in very large databases, Proceedings *of the ACM SIGMOD Conference on Management of data*, p. 207-216.

[15] Jeffrey Xu Yu, Michael, K., Ng, Joshua Zhexue Huang, (2001). Patterns Discovery Based on Time-Series Decomposition, *Advances in Knowledge Discovery and Data Mining: 5th Pacific-Asia Conference, PAKDD* 2001 Hong Kong, China, April.

[16] Vijayalakshmi, M., Menezes, B. (2010). Using Data Mining to Identify a Consistent Set of Experts for Times Series Sales Forecasting, *AIA* 2010, paperId=37731, http://www.actapress.com/Abstract.aspx?.

[17] Vijayalakshmi M., Designing a Frequent Pattern Mining Based Sales Forecast Engine, paper submitted to *OPSEARCH - Journal of the Operations Research Society of India*.

[18] http://empslocal.ex.ac.uk/people/staff/dbs202/cat/stats/corr.html

[19] Why do we sometimes get nonsense correlations with time series?, http://www.math.mcgill.ca/dstephens/OldCourses/204-2008/Handouts/Yule1926.pdf

[20] Viger Philippe Fournier, SPMF- A Sequential Patter Mining Framework, http://www.philippe-fournier-viger.com/spmf/

[21] Warren Liao, T. (2005). Clustering of time series data-a survey. *Pattern Recogn.* 38 (11) 1857-1874, Nov. DOI= http://dx.doi.org/10.1016/j.patcog. 2005.01.025

[22] MJ Zaki, CJ Hsiao, (1991). CHARM: An efficient algorithm for closed association rule mining, 2*nd SIAM International Conf. on Data Mining*, 457-473.

[23] Economagic.com: Economic time series page. http://www.economagic.com/