**Abstract**

Millions of products come from multiple vendors into the inventory. Most vendors source their inventory based on market needs. So market changes can be identified by keeping track of inventory changes over time.

We have analyzed a model that analyzes the products added by vendors every day/week and suggests which attributes are trending for each vendor, which attributes are trending in the market, which vendors are trend setters, and which attributes are going to be in trend in near future. We have investigated two different ways of finding the trendiness of vendors for attributes and trying to predict the future trends from them, namely by a formula and by regression. The results have shown that the two ways do not predict correctly.

**1 Introduction**

We have millions of products coming from multiple vendors. Each product is classified into various categories based on certain attributes like Color, Material, Pattern, Length, etc.

Most vendors source their inventory based on market needs so in essence our inventory is a snapshot of what market needs. But we might be having a lot of old inventory which skews the data if analyzed in real time. So we need a system which keep track of inventory changes over time and can clearly identify market changes.

In this paper, we use the term attribute for different attributes like Color, Length, etc. and attribute value for different values of these attributes like red, blue, multicolor, etc. for attribute color.

For each vendor, the trending attribute values are determined by giving each attribute value a Rating. Higher the Rating of an attribute value for a vendor, more is that attribute value trending for the vendor. The model uses time decay for evaluating these Ratings. We have explored different ways of normalizing these Ratings for comparing values of different attributes.

The trending attribute values for the whole market are determined by giving each attribute value an OverallRating. Higher the OverallRating, more is that attribute value trending. To identify the vendors that identify the trend in an attribute earlier, the model gives each vendor-attribute pair a trendiness value. Higher the trendiness value of a pair, more likely is the vendor to identify the trend earlier. To predict the trending attribute values in the near future, the model uses the Ratings on a day and takes a weighted sum of them for each vendor weighted by the trendiness values for the vendor-attribute pairs.

**2 Details of approach**

**2.1 Ratings of attribute values**

Ratings of attribute values for each vendor can be calculated by three ways:

1. No. of products in the inventory with given attribute value for given vendor.
2. No. of products recently added in the inventory (a week or a month before) with given attribute value for given vendor.
3. No. of products added with a time decay of the age with given attribute value for given vendor.

Since there can be a lot of old inventory that skews the data, it is not sensible to simply count all the inventory, so option (i) is rejected. But counting only the new inventory is not sufficient because many vendors do not add the inventory very often and their trending attribute values would be of products added long back. So, option (ii) is rejected. The time decay logic is accepted and fits the given data too.

So, currently the model operates as follows. After the Ratings are initialized on a day, then every subsequent day new Ratings are calculated by giving weightage 1 to the newly added products and a weightage<1 (decay factor) to the Rating on the previous day.

**2.2 Ratings of values of different attributes**

To compare the Ratings of attribute values of different attributes, normalization is needed because they are not on same scale. For example, the Ratings may be 100 for 10 different colors but may be 500 for two different lengths, but that does not mean that the two lengths are more in trend than the 10 colors.

We considered three different ways of normalization:-

1. (x-mean),
2. (x-mean)/(standard deviation) and
3. (x/mean).

(x-mean) was discarded because it would show the trend values of (20, 80) and (120, 180) as similar but the first set of attributes shows a more marked difference in trend values than the second set. (x-mean)/(standard deviation) was also discarded because it would also fail in the same example. So, (x/mean) has been selected as the way of normalization as it scales the values to a similar level while preserving the variance.

**2.3 OverallRatings of attribute values**

For calculating the vendor independent ratings of attribute values, there are two different ways:-

1. The same logic as ratings, as time decay with mean normalization.
2. Average of ratings of vendors, weighted by the means for attributes of the vendors.

It was seen from the data that the method (i) gave OverallRatings on a quite different scale than the Ratings, and the OverallRatings of different attributes were on a different scale. On the other hand, the method (ii) produced OverallRatings very comparable to the Ratings and comparable for different attributes. So, the method (ii) has been chosen for the model.

**2.4 Trendiness of vendors for attributes**

To find the vendors who could identify trends early and act accordingly, we considered that whatever attribute values such vendors would make trending, later on the other vendors would follow. So, we needed to identify those vendors whose trending attributes the market would follow after a few days or weeks. So, whatever trending values those vendors had for attribute values, the market would approach those values after some weeks. We also considered another factor, such vendors would not generally stay much away from the current trends in market. So, we came up with the formula which would give the trend values of the vendors for a given attribute value. We took the mean over all attribute values for a given attribute to get trend values of vendors for given attributes. Now, we had the data of trendy vendors but we did not have the right data to verify our prediction.

**2.5 Future trends of attribute values**

Then we decided to check the trendiness of vendors by predicting the future data given the current data and the trendiness of vendors for different attributes. Then we discovered a great mismatch between trend values for vendors and trend values for market. We changed the formula of trend values for market as an average of the trend values for vendors weighted with the means of trend values for attributes. This not only corrected the mismatch, but also did not adversely affect the comparison of trend values of attribute values of different attributes in the market. Now, the model is running on this formula.

So, to check the trendiness of vendors, we thought that we can take an average of the predictions of the vendors, weighted by their trendiness values. We considered the predictions of the vendors as the difference between their trend values and the trend values of the market. So, the formula we came up with was . We used it to see how close it was to the real change in data till tomorrow. However, we observed that it did not agree well with the data next day.

Then we came up with the idea to get the right trendiness values of vendor-attribute pairs by training the data to predict the future data with trendiness values as parameters. Here comes the idea of regression. We have data of every day, and we train the model to predict the trend values of the market the next day given the trend values of the vendors this day. We can then test it on testing data and get the results. When we tested this on training data, we saw much better correlation than the previous model.

On the other hand, when we tested on testing data, we could not find much correlation, which means that the model was working well with training set but failed with the testing data, and hence there seemed to be the problem of overfitting of data.

**3 Conclusions**

We have seen that prediction of future trends by the formula does not give right results. We have also seen that regression causes the model to give right results for the training data but not for the testing data.

**Acknowledgements**

We thank Adoro for the access to the product data.