Product Grouping Methods for Markdown Pricing

Gautam Chauhan [1]   Katie Howgate [2]

[1]University College London (UCL)   [2]STOR-i Centre for Doctoral Training

Motivation

A markdown is a permanent reduction in price which happens when a product in a store reaches the end of its lifetime. Some important examples of this include:

- Food in a supermarket approaching its expiry date
- Obsolete technology
- Out of fashion or out of season clothing

The goal of markdown pricing is to choose an optimal series of price reductions which maximises revenue from the unsold inventory. To do this, we need to estimate demand for the product at each possible sale price - this is called a Price Response Function.

One key issue is that markdown data is normally only available for a small number of closely-spaced prices, which makes demand modelling challenging. A possible solution is to group products by their price response functions in a normal sales period. The goal of this project is to investigate clustering methods which could be used for this product grouping (using simulated data).

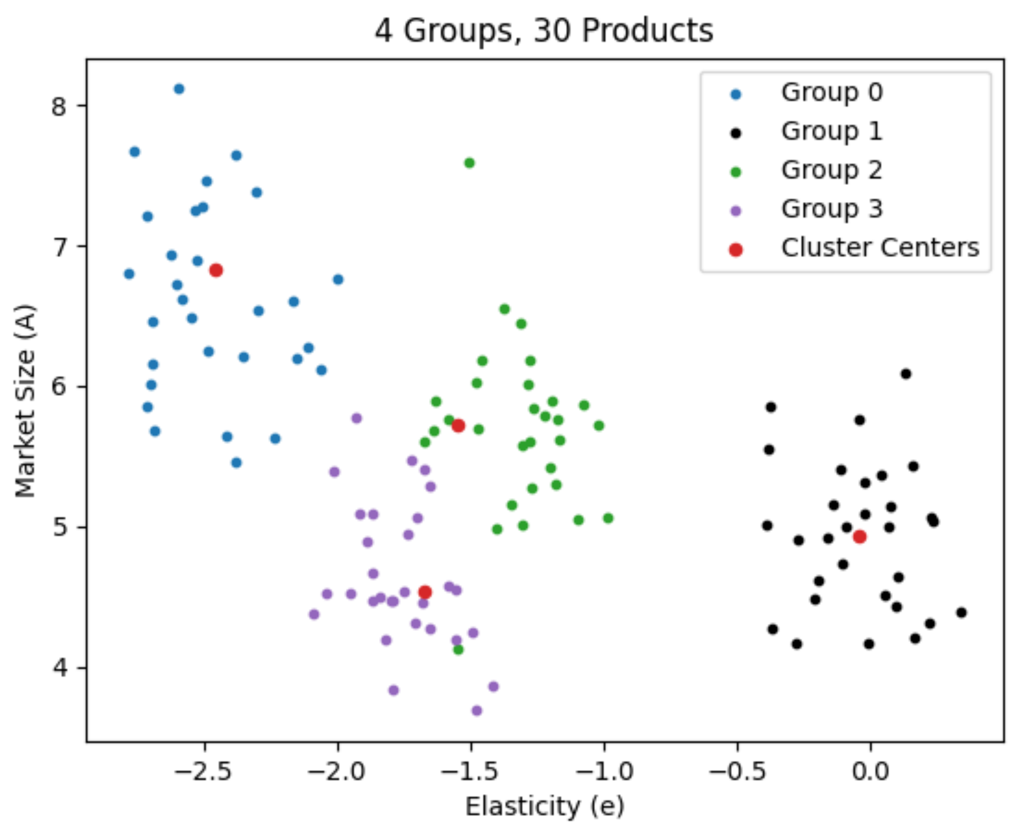
K-Means Clustering

K-Means is a clustering method which partitions a set of observations into a predefined number of groups (K). The mean of the observations in each cluster is called a centroid - the objective is to choose their positions to minimise the within-cluster sum of squares:

$$\text{arg min} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} ||\mathbf{x} - \mu_{\mathbf{i}}||^2$$

To apply K-Means to the product grouping problem, we use this process:

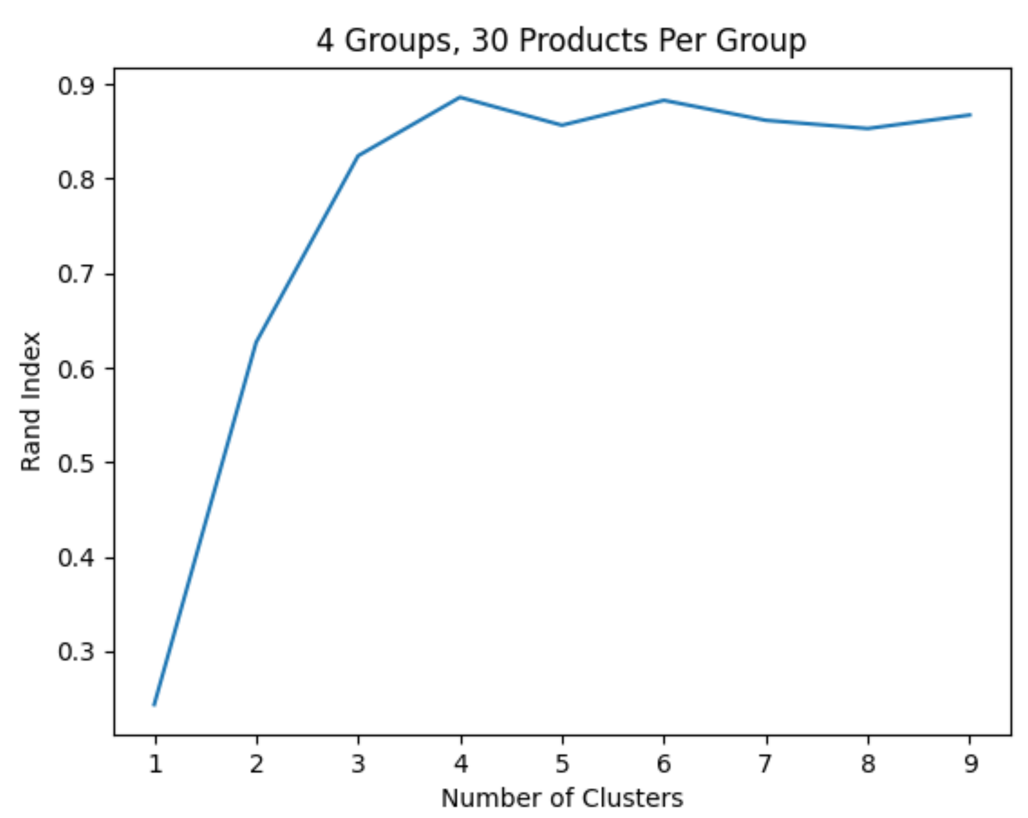
1. Suppose the price and demand data can be modelled by $d(p) = Ap^e$. This means $log(d) = log(A) + elog(p)$
2. For each product, perform linear regression on log(d) and log(p): the slope and intercept can be used to estimate e and A.
3. Apply the K-Means Clustering Algorithm to the (e, A) data:



Rand Index is a metric which takes each pair of observations and compares the way they have been grouped to the way they are in the true clusters (TP = True Positive, TN = True Negative, FP = False Positive and FN = False Negative):

$$\text{Rand Index} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$$

A value closer to 1 indicates better agreement with the true clusters, so we can use the value of the Rand Index to work out the optimal number of clusters:



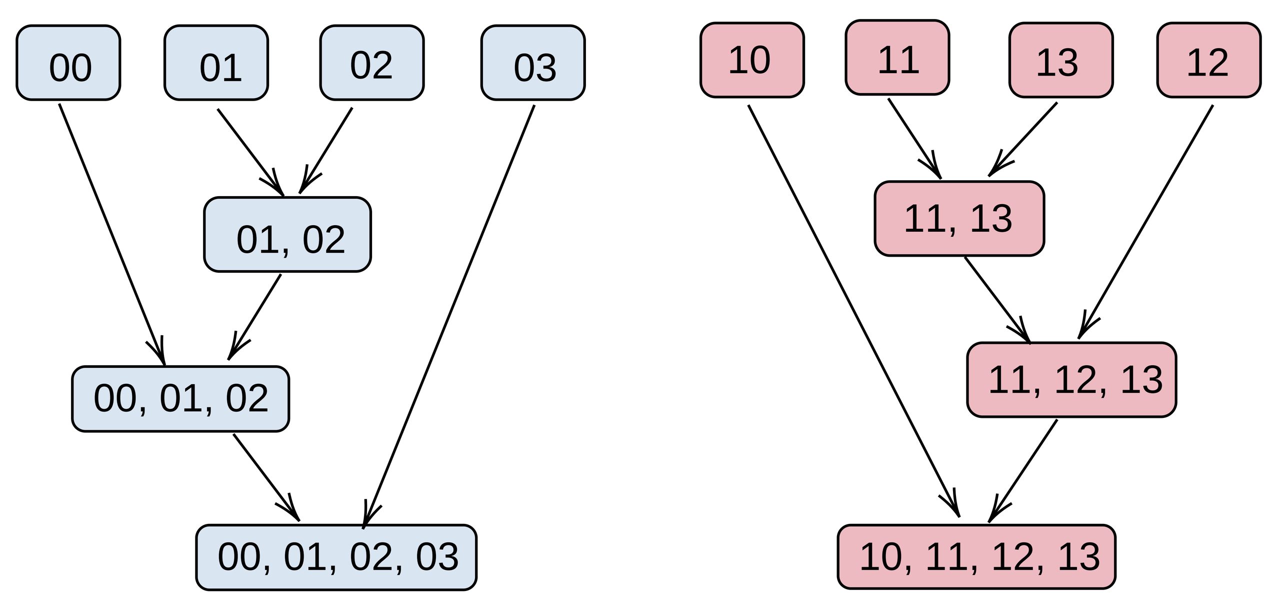
Bayesian Hierarchical Clustering

Bayesian Hierarchical Clustering is a method which places the products into separate clusters and fits the price and demand data in each cluster using Gaussian Process Regression. Pairs of clusters are then merged until the total log marginal-likelihood is maximised.

The algorithm used is as follows:

- Assign each product to its own cluster
- Fit the price and demand data in each cluster using Gaussian Process Regression and calculate the total log marginal-likelihood
- While there is an improvement:
  - Merge each pair of clusters, fit the price and demand data using Gaussian Process Regression and find the change in log marginal-likelihood
  - Find the merge which produces the biggest increase in log marginal-likelihood
  - If this is positive, accept this merge and repeat the process.
  - Else, the log marginal-likelihood has been maximised and we have found the optimal clustering, so end the process.

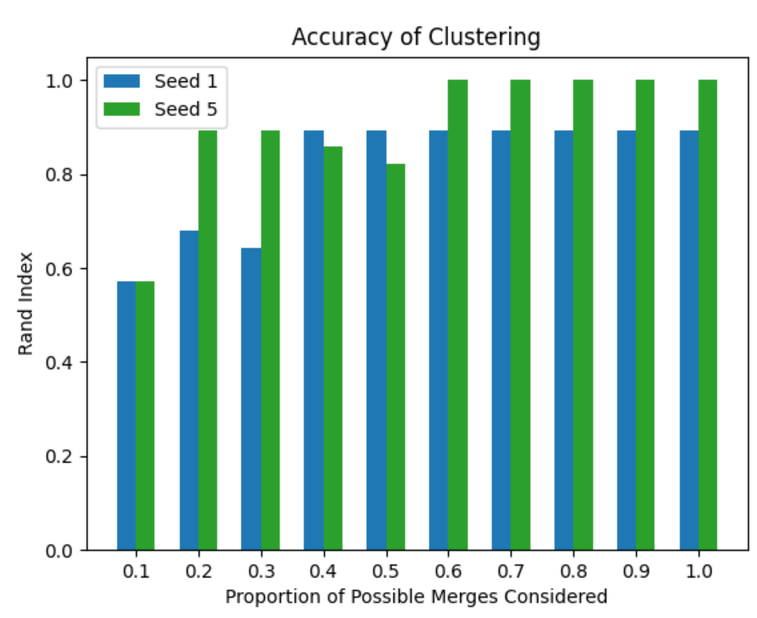
The dendrogram below illustrates the clustering process when we apply the algorithm to 2 groups with 4 products in each group:

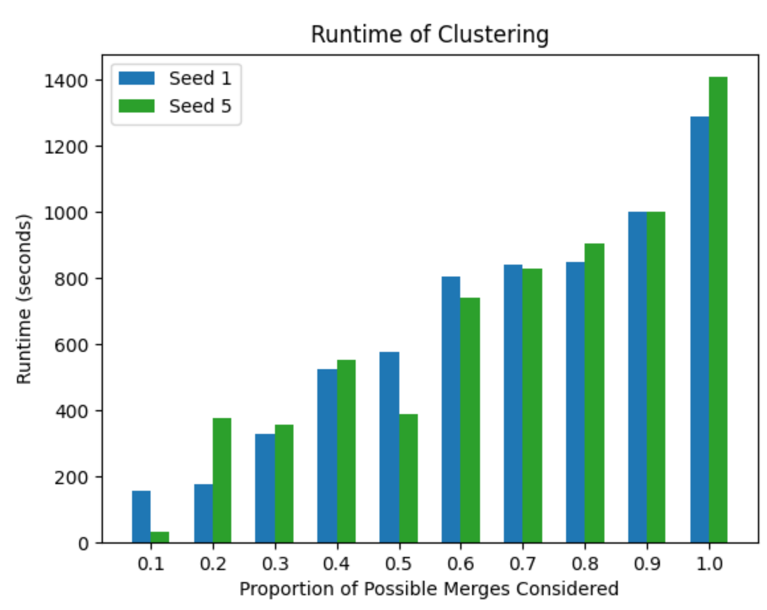


The runtime for this clustering algorithm is $\mathcal{O}(D^3N^3)$, where D is the number of (price, demand) observations for each product and N is the number of products (and therefore starting number of clusters).

A potential way to speed the process up and obtain an approximate solution is by choosing a random sample of cluster pairs to try merging at each step of the process, instead of trying all possible pairs.

The graphs below show how the runtime and accuracy (as measured by the Rand Index) vary with the proportion of possible merges considered:





Conclusions

Method	K-Means Clustering	Bayesian Hierarchical Clustering
Advantages	Good Time Complexity	Uses non-parametric model to relate price and demand Calculates optimal number of clusters
Limitations	Requires parametric model to relate price and demand Choosing optimal number of clusters is hard and subjective	Poor Time Complexity: $O(D^3N^3)$

References

[1] Christopher K. Williams Carl Edward Rasmussen. *Gaussian Processes for Machine Learning*. Cambridge: The MIT Press, 2019.

[2] Hensman et al. Hierarchical Bayesian modelling of gene expression time series across irregularly sampled replicates and clusters. *BMC Bioinformatics*, 2013.

[3] Manning et al. *Introduction to Information Retrieval*. Cambridge: Cambridge University Press, 2008.

[4] Robert L. Phillips. *Pricing and revenue optimisation*. Stanford, California: Stanford Business Books, 2021.

[5] Thomas Pinder and Daniel Dodd. GPJax: A Gaussian Process Framework in JAX. *Journal of Open Source Software*, 7(75):4455, 2022.