ECONOMIC STATISTICS CENTRE OF EXCELLENCE

A COLLABORATION WITH Office for National Statistics

Applying Machine Learning to Detect Outliers in Alternative Data Sources. *A universal methodology framework for scanner and web-scraped data sources*

Xuxin Mao, Janine Boshoff, Hande Küçük and Garry Young

ESCoE Technical Report TR-12

November 2021

ISSN 2515-4664

TECHNICAL REPORT

Applying machine learning to detect outliers in alternative data sources. A universal methodology framework for scanner and web-scraped data sources
Xuxin Mao, Janine Boshoff, Hande Küçük and Garry Young
ESCoE Technical Report 12
November 2021

Abstract

This research explores new ways of applying machine learning to detect outliers in alternative price data resources such as web-scraped data and scanner data sources.Based on text vectorisation and clustering methods, we build a universal methodology framework which identifies outliers in both data sources. We provide a unique way of conducting goods classification and outlier detection. Using Density-based spatial clustering of applications with noise (DBSCAN), we can provide two layers of outlier detection for both scanner data and web-scraped data. For web-scraped data we provide a method to classify text information and identify clusters of products.

The framework allows us to efficiently detect outliers and explore abnormal price changes that may be omitted by the current practices in line with the 2019 Consumer Prices Indices Manual 2019. Our methodology also provides a good foundation for building better measurement of consumer prices with standard time series data transformed from alternative data sources.

Keywords: Consumer Price Index, Text Density based clustering, Machine Learning, Outlier Detection, Scanner Data, Web-scraped Data

JEL classification: C43, E31

Xuxin Mao, National Institute of Economic and Social Research, x.mao@niesr.ac.uk

Published by:
Economic Statistics Centre of Excellence
National Institute of Economic and Social Research
2 Dean Trench St
London SW1P 3HE
United Kingdom
www.escoe.ac.uk

ESCoE Technical Reports describe the methodology and research in progress by the author(s) and are published to elicit comments and to further debate. Any views expressed are solely those of the author(s) and so cannot be taken to represent those of the Economic Statistics Centre of Excellence (ESCoE), its partner institutions or the Office for National Statistics (ONS).

ECONOMIC STATISTICS CENTRE OF EXCELLENCE

A COLLABORATION WITH Office for National Statistics

Applying Machine Learning to Detect Outliers in Alternative Data Sources

A Universal Methodology Framework for Scanner and Web-Scraping Data Sources

Xuxin Mao[1], Janine Boshoff, Hande Kucuk, and Garry Young

Nov 2021

This research has been funded by the Office for National Statistics as part of the research programme of the Economic Statistics Centre of Excellence (ESCoE). Methodology report submitted as part of ESCoE Project 1.11: Incorporating alternative data sources into price indices. The authors thank ONS colleagues, especially continuous research support from Sanjiv Mahajan and Helen Sands. Please do not quote without permission from the authors.

[1] Corresponding author. x.mao@niesr.ac.uk Xuxin is a Research Associate at ESCoE, Principal Economist at NIESR, and Visiting Fellow at LSE European Institute.

Abstract

This research explores new ways of applying machine learning to detect outliers in alternative price data resources such as web-scraping data and scanner data sources. Based on text vectorisation and clustering methods, we build a universal methodology framework which identifies outliers in both data sources. We provide a unique way of conducting goods classification and outlier detection. Using Density-based spatial clustering of applications with noise (DBSCAN), we can provide two layers of outlier detection for both scanner data and web-scraping data. For web-scraping data we provide a method to classify text information and identify clusters of products.

The framework allows us to efficiently detect outliers and explore abnormal price changes that may be omitted by the current practices in line with the 2019 Consumer Prices Indices Manual 2019. Our methodology also provides a good foundation for building better measurement of consumer prices with standard time series data transformed from alternative data sources.

JEL codes: C43, E31.

Key Words: Consumer Price Index, Text Density based clustering, Machine Learning, Outlier Detection, Scanner Data, Web-Scraping Data.

1. Introduction

Consumer Price Indices (CPI) have typically been compiled using data collected by price collectors, either remotely or in-store. Now that web-scraping and point of sale scanner data are available they provide excellent sources of price data to be used by the Office for National Statistics (ONS, 2019 and 2021) and its international counterparts (Boshoff, Mao and Young, 2020). These sources are known as “alternative” data since the information is collected for purposes outside the remit of statistical agencies and the production of official statistics. Before incorporating these data, accurate outlier detection is extremely important to identify and invalidate price movements that differ significantly from the norm for a particular item.

Outliers are data points that differ significantly from other observations (Grubbs, 1969; pp 89, Maddala, 1992). A price observation that is significantly different from the average price for that particular good is considered an outlier. It is important to detect the erroneous information from the dataset because it could result in calculated inflation appearing much higher or lower than it really is. Outlier detection, followed by further ONS outlier scrutinisation practices, provide us with the best chance of identifying erroneous values that may have a detrimental effect on the price index.

To detect outliers requires us understanding one key characteristic of the dataset, i.e., whether it is parametric or non-parametric. Parametric data comes from a population that can be adequately modelled by a probability distribution that has a fixed set of parameters. Conversely a non-parametric model makes no assumptions about a parametric distribution.

Currently, most statistical agencies use parametric outlier detection methods such as statistical profiling, i.e., creating upper and lower bound cut-off points by adding and subtracting a fixed number of standard deviations from a mean or median. The ONS uses a combination of manually validating prices exceeding a ten-fold growth relative to the January comparison month and a modified Tukey method where outliers are marked in two mean-split cohorts when a price change exceeds 2.5 standard deviations from the central mean (ONS, 2019).

While machine learning methods have been used for anomaly detection from various data sources in many different fields, they have not been adopted for alternative price data sources. Most practices of official statistical agencies focus on the following areas for price statistics: exclusion of clearance prices and relevant aggregation to reduce impact of outliers; threshold calculation based on percentage change and confidence intervals; adjusted boxplot based on exponential models; threshold calculation, outlier filtering and two dumping filters (Boshoff, Mao and Young, 2020).

Due to their large volume and unique nature, we do not always know the distribution of data from alternative sources, requiring non-parametric approaches to detect outliers. Statistical profiling is not suited to deal with the text information and complex product identification that characterises alternative data sources. While it is difficult to isolate the impact of changes in pricing, features, or packaging on sales quantity in the case of scanner data, the complicated and changeable nature of webpage structures can affect the reliability of web-scraping data. In line with Charlton's (2020) review of some outlier detection methods for alternative prices data, this paper provides an AI-based nonparametric framework to address outlier detection issues.

The methodology and algorithm we developed classifies products based on descriptive text information, detect the products' abnormal price changes and produce standard time series data for further analysis. While product categories can be provided in scanner data, for web-scraping data we develop a text density-based clustering method to classify goods based on their key features and filter them for the representative items while dropping any attribute descriptions that are not useful for further data analysis.

The methodology framework makes contributions on two aspects. Firstly, it provides a way of identifying products based on descriptive text information, which can be used to build a standard price index. Secondly, it can detect outliers that vary from current mainstream statistical profiling for further scrutiny and analysis. Given there is a risk of false positives, which are dependent on parameter setup, we focus on using the outlier detection framework to narrow our search for erroneous values.

2. Data Description

This section describes the two sets of data we use for outlier detection, namely scanner data and web-scraping data.

The data are collected by retailers at the point of sale, providing statistical offices with enhanced product, geographic and temporal coverage as well as significantly more information on the number and type of products sold, reflecting changing consumer spending patterns more accurately. We use scanner price data from Dominick's Finer Food (DFF), a large Midwestern US supermarket chain, operating about 130 stores in the Chicago Metro area. Dominick's price data contain 98,691,750 weekly price observations for up to 399 weeks from 14 September 1989 (week 1) to 05 January 1997 (week 399) for 18,037 different products (SKU's) in 29 categories. These are actual transaction prices that consumers have paid each week, as recorded by the chain's scanners at the checkout cash registers.[2]

The dataset includes three main types of data: Customer count file, store-specific demographics, and category-specific data. The customer count file captures store-

[2] Dominick's data can be downloaded from the University of Chicago Business School's website: https://www.chicagobooth.edu/research/kilts/datasets/dominicks

specific daily data including in-store traffic, the total dollar sales and coupons redeemed for each of the DFF department. The store-specific demographics data are based on the U.S. (1990) census data for the Chicago metropolitan area processed by Market Metrics to generate store-specific demographic data.

While the first two categories of data provide good background information on the goods prices, the category-specific data provide more detailed micro-level data to examine price changes. The category-specific data include files describing categories with unique Universal Product Codes (UPC), and movement files containing weekly store-level sales information for each UPC in a category.

We use the beer category for this analysis that contains 3,846,701 observations of 790 forms of beer sold at DFF (bottles, cans ...). The registration of category starts in Week 91 and ends in Week 399. The category itself covers a lot of information to be carefully examined for the potential missing values, additional price ranges, and more extreme price changes.

We intend to find two different types of outliers: outliers across shops and over periods. Although Dominick's prices are set on a chain-wide basis at the company headquarters, there are price variations across the stores. This may raise the question whether some features related to the shops play some role in price changes.[3]

Web-scraping data involve collecting prices and other related product information automatically from online websites. These data provide a wealth of additional product information about online prices, such as product descriptions, which require extra effects to cluster and analyse for outlier detection.

Our research uses web-scraping DVDs data as our data sample with 120,728 entries including the product ID, date, price, location, and attribute information. The attributes include number of discs, boxset, classification, region, language, brand, product description, etc. While streaming is getting popular, DVDs are still incredibly useful and widely consumed. In the meantime, the algorithm we build for DVD text data can be readily used for many other important CPI related products based on their similar web-scraping texts.

There are three layers of outlier problems in web-scraping data. We need to first detect outliers when we cluster the web-scraping data into specific product groups. Unlike scanner data, the web-scraping data do not have unique identification information for specific goods. We rely on their descriptive information such as characteristics, prices, and dates to classify the goods into separate groups and identify them based on text clustering. Besides making sure that only suitable entries can be included, and proper

[3] When scaled up, our algorithm can still detect outliers fast from either scanner or web-scraping data.

product clusters are determined, we have two layers of outlier detection similar to the methodology developed for scanner data. We need to detect outliers from the similar products within the same month. We then use the processed prices as time series observations which will be subject to outlier detection and validation.

3. Machine Learning Models and Algorithms

This section presents the machine learning tools and methods that we use for outlier detection, covering natural language processing and clustering algorithms.

3.1 Doc2Vec Model for Natural Language Processing

Applying machine learning to natural language faces one major hurdle: algorithms usually deal with numbers while natural language is text. We therefore need to transform that descriptive text into vector numbers, otherwise known as text vectorisation. It is a fundamental step in the process of machine learning for analysing data, and different vectorisation algorithms may differ in their efficiency or accuracy, which means that we need to choose the best one that suits our purposes.

Numeric representation of text documents is a challenging task in machine learning. Such a representation may be used for many purposes, including document retrieval, web search, spam filtering, topic modelling, etc. There are several methods that could be used for text vectorisation, such as Word2Vec (Mikolov et al., 2013), FastText (Bojanowski et al., 2016; Joulin et. al., 2016), BERT (Devlin et al., 2018), and Doc2Vec (Le and Mikolov, 2014).

Word2Vec is a fundamental natural language processing approach which can be used to derive vectors with adjustable length. While FastText is essentially the sum of weighted average of word vectors derived by Word2Vec, BERT fixes the length of all vectors to be 768 dimensions, which requires a lot of training time which may prolong calculation and affect efficiency.

Meanwhile, Word2Vec, FastText and BERT are methods specially designed to deal with semantic information of words, they may not be a good fit for our purpose as we aim to derive information directly from a complete text document. Compared with other methods, Doc2Vec is the preferred choice as it can generate vectors with flexible length like Word2Vec and FastText and derive results with more efficiency and less time than BERT when model training time is considered. Meanwhile, it provides more accurate information than other methods as it is specially tailored for vectorisation of complete texts. Accordingly, for web-scraping data documents with descriptive information, we propose to use a Doc2Vec method to transform text information into vector format for further outlier detection.

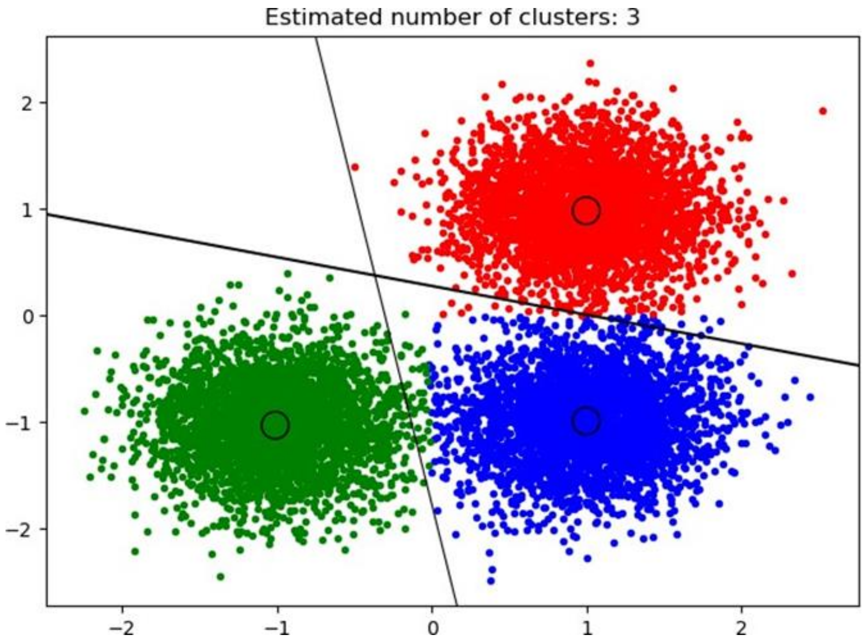
3.2 DBSCAN Algorithm as Outlier Detection Method

with derived vector information, we can then adopt one clustering method to detect outliers. A well-known data clustering algorithm that is commonly used in data mining and machine learning, the density-based spatial clustering of applications with noise (DBSCAN), is used in our research.

Based on a set of points, e.g., in a bidimensional space as exemplified in Figure 1, a DBSCAN model groups together points that are close to each other based on a distance measurement (usually Euclidean distance) and a minimum number of points. It also marks as outliers the points that are in low-density regions.

The idea behind DBSCAN is relatively simple. If two points are within a certain distance of each other, they are considered as neighbours. If there are enough neighbours within a region, that region is considered as "dense". This gives us two controllable parameters with which to be able to build our clusters: specifically, epsilon as the distance at which points can be considered as neighbours, and minPoints as the number of points in a region for it to be considered dense.

Figure 1: DBSCAN Cluster Estimation



Note: Authors' own calculation based on random text documents of Web-Scraping data

To choose good parameters, we need to understand how they are used and have at least a basic previous knowledge about the dataset that is being examined. If a chosen epsilon value is too small, a large part of the data will not be clustered. It will be considered as outliers because it does not satisfy the number of points to create a dense region. On the other hand, if the value chosen is too high, clusters will merge, and most objects will be in the same cluster. The epsilons should be chosen based on the distance of the dataset, but in general small epsilon values are preferable.

When selecting the minimum number of points that we need in an area, for that area to be considered data rich (which is the essence of it being dense), there is no canonical formula. Careful hand crafting is required based on the application. If we choose a threshold which is too small, more regions can appear dense where the values' relationship is unclear. Conversely, if the threshold is too large a value can create areas which should be used but are either separated or not considered dense.

This figure is included for illustrative purposes. The vectors we work with in reality may be as high as 1000 dimensional. However, the figure allows us to describe two different types of point within each region. Suppose we draw a ball around every data point in a region and count the number of points that fall inside that ball. If the balls are drawn so that the radius is the same as the distance, we have chosen what defines neighbours, and we can have one of two situations

- Internal point – these points are those for which the ball is itself considered a dense region (so it contains at least as many points as we have used to define density)

- Border point – these are points for which the ball around it contains strictly fewer points than the density threshold.

We can then detect outliers as sample points which are not core points or border points. Figure 1 presents a simple 2-dimentional cluster estimation illustration of DBSCAN with random sample texts from our web-scraping data set. We can find that DBSCAN can generate three different clusters, each of which was surrounded by a core point in the center.

We can cluster our information into four areas, three of which have distinctive colours to represent unique goods. While core points are the densely populated area in each colour, border points are sparsely distributed points approaching the border lines. The outliers are the points that lie inside the areas dominated by points with other colours. For example, the green points inside the bottom right area that is dominated by blue points can be detected as outliers.

DBSCAN is a flexible non-parametric method that deals well with clusters of different densities like the web-scraping data. It does not require a specific setting on normal distribution or any other specific distribution, with relatively few assumptions and hyper-parameters to tune. DBSCAN is easy and fast to calculate, and available in popular computing packages such as Python's scikit learn. While applicable to alternative data sources, DBSCAN can be used for detecting outliers from standard data like cross section or time series data.

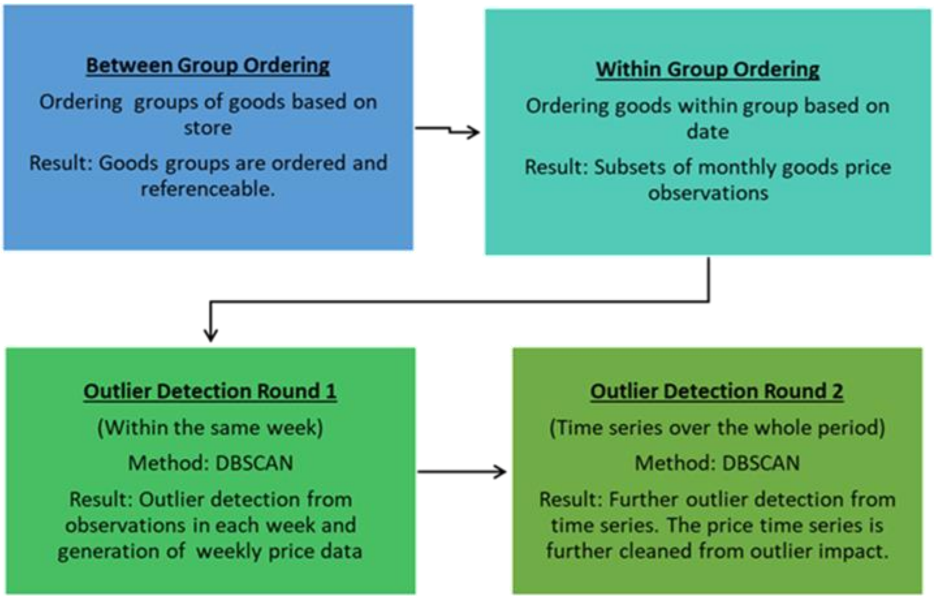
Furthermore, compared with other popular methods like the Isolation Forest (Liu et al., 2008), DBSCAN deals flexibly with data features, such as trend changes or breaks, that could increase the risk of identifying false positives. The core idea of DBSCAN is to continuously determine the borderline points to be included in the cluster of core points instead of outliers. Continuous changes are more likely to stay in the time series based on DBSCAN while other methods are more likely to be affected by specific price changes in the previous periods.

To better use a DBSCAN algorithm, we train it with clean data. Therefore, an assumption is made that there are no misclassifications or other errors during the training data period, which is usually the first month of collection. DBSCAN may have a bias when there are effects of churn in prices data. To overcome this problem, it is advisable to have a rolling training period (Mayhew and Clews, 2016). Meanwhile, the choice of epsilon values is determined on a product-by-product basis. For example, for products with prices that are traditionally stable such as milk, an epsilon of 1 would not make sense on a 60p product with little variance, whilst on DVDs it could be far more sensible. While we use indicative numbers in the paper as epsilon values, a per-good threshold recommendation engine defined by ONS experts will be useful for future DBSCAN analyses.

4. Applying Machine Learning for Scanner Data Outlier Detection

The machine learning procedures for outlier detection from both scanner data and web-scraping data involve the following Python packages: jieba, genism, sklearn, numpy and matplotlib. We first apply the methodological framework to scanner data, since its structural nature allows for easier examination of its outliers than web-scraping data. The outlier detection architecture is illustrated by Figure 2.

Figure 2: Architecture of Outlier Detection for Scanner Data


Between Group Ordering
Ordering groups of goods based on store
Result: Goods groups are ordered and referenceable.
Within Group Ordering
Ordering goods within group based on date
Result: Subsets of monthly goods price observations
Outlier Detection Round 1
(Within the same week)
Method: DBSCAN
Result: Outlier detection from observations in each week and generation of weekly price data
Outlier Detection Round 2
(Time series over the whole period)
Method: DBSCAN
Result: Further outlier detection from time series. The price time series is further cleaned from outlier impact.

4.1. Between Group Ordering of Goods

We first access the beer data and order them based on the unique identifiers (UPC). This information is then listed over time and across different stores.

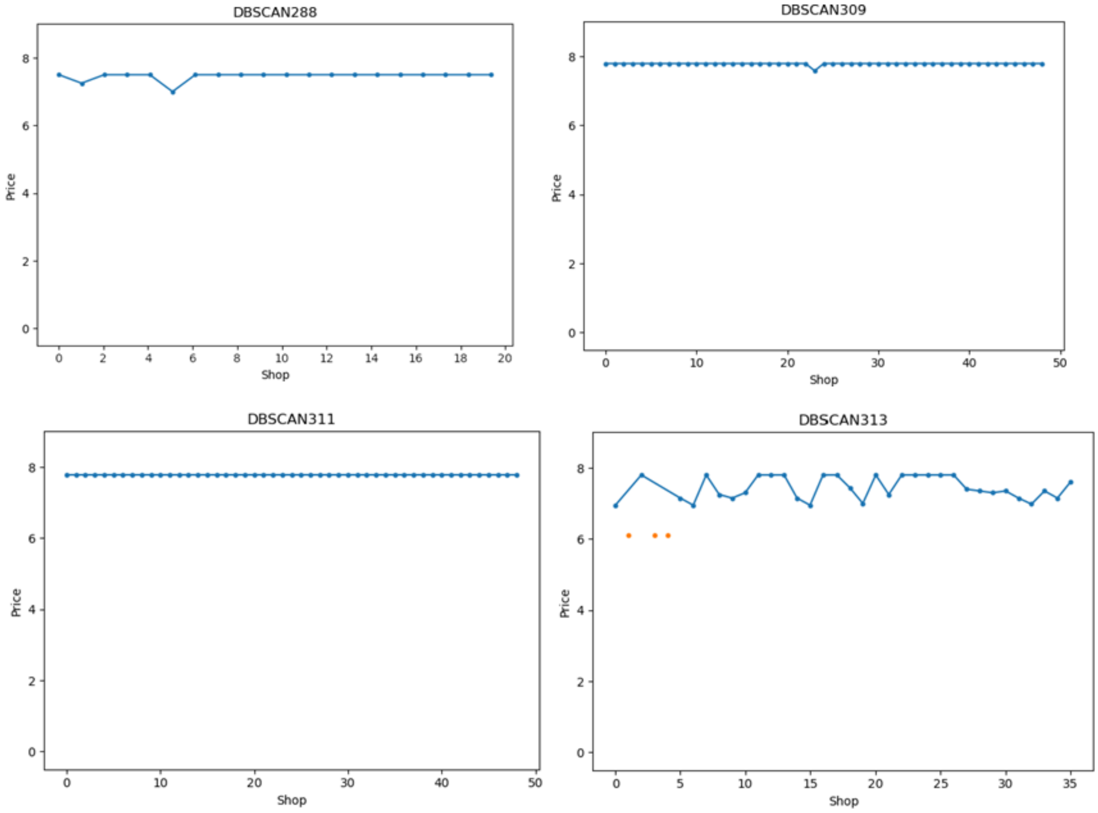
4.2 Within Group Ordering of Goods

We then sort the goods within the same UPC group based on the date information. This step gives us subsets of beer price data within the same week. The two steps provide a good data foundation for further machine learning practices.

4.3 First Round Outlier Detection for Scanner Data

In this section we pick up the group of beer goods with UPC number 1820000613, and use 4 weeks of data (Week 288, Week 309, Week 311, Week 313) to illustrate how to use DBSCAN algorithm for detecting outliers from price data within the same week.

Figure 3: Within Group Outlier Detection Results for Beers in Same Periods[4]



[4] The names of DBSCAN288, DBSCAN309, DBSCAN311 and DBSCAN 313 indicate the observation periods of week 288, week 309, week 311 and week 313 respectively. The X-axis represents the shop ID while the Y-axis indicates the corresponding price of the beer. Outliers are indicated in orange.

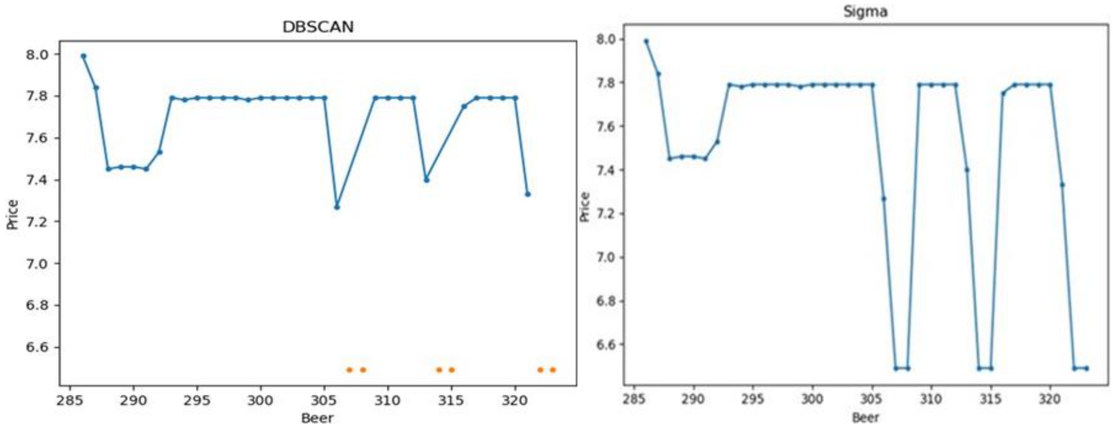
With epsilon value of 0.5 and MinPoint=3, we conduct DBSCAN procedures to identify outliers. Based on the normal distribution of price data, a 2.5 standard deviation is approximately $0.5, and as such we select the epsilon value to be 0.5. While we focus on price levels in this study based on assumptions of normal distribution and standard deviations, the similar approach can be readily adopted for price changes if we have information on threshold values in terms of abnormal price changes. Figure 3 shows that in Week 313 there are 3 outliers out of 37 shops selling the specific beer. All the outliers happen to represent noticeable price level changes and so warrant further scrutiny.

In Weeks 288, 309 and 311, there are no outliers detected in the beer group across all the shops. We can take the average of the non-outlier price points to produce a point price observation for the week, which then can generate a time series with weekly frequency for further analysis.

4.4 Second Round Outlier Detection for Scanner Data

With the price time series derived from Section 4.3, we continue to filter abnormal prices and establish a more stationary time series. For this step, we can use DBSCAN and compare the results with a standard outlier detection method adopted by the official statistics institutions.

Figure 4: Between Group Outlier Detection Results for Beers Over Periods



DBSCAN can detect the most significant abnormal price level changes which can be useful to improve consumption price measurement and CPI construction. Shown in Figure 4, the specific beer was available for sale over the weeks 285-323. Based on DBSCAN (epsilon=0.5 and MinPoint=3), we can find 6 outliers from time series data. All the outliers reflected a sharp price change of more than half dollar, which are identified for further inspection. By comparison, using the Tukey[5] method currently

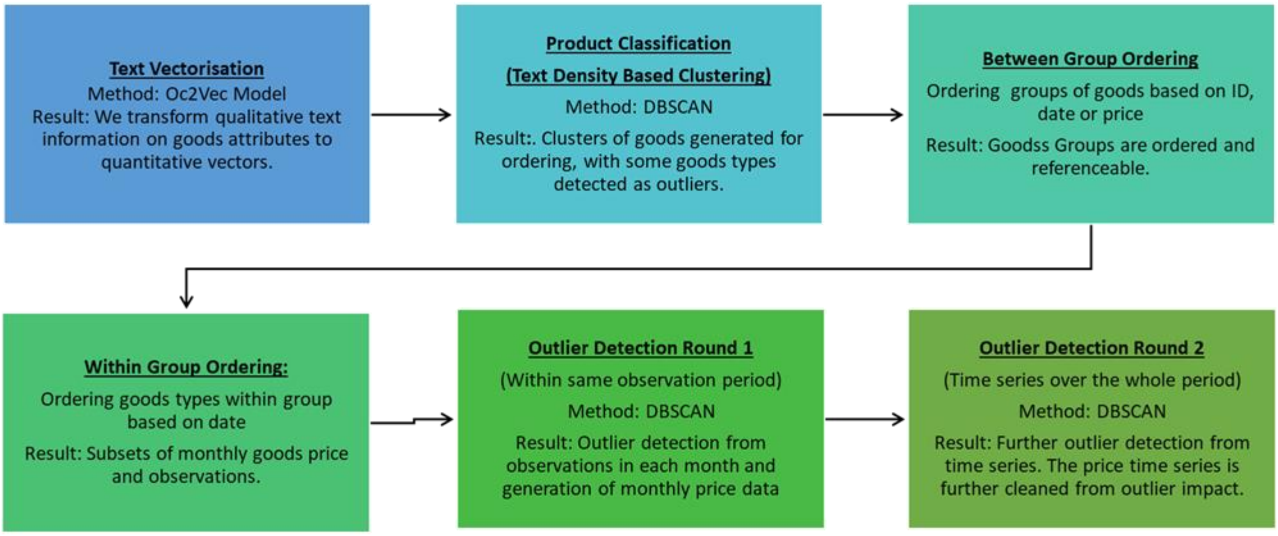
[5] The Tukey method defines the limit as the trimmed mean plus (or minus) 2.5 times the difference between the trimmed mean and the upper (or lower) midmean.

employed for ONS outlier detection (ONS 2019), we do not detect any outliers (the right-hand chart of Figure 4) for further examination.

5. Applying Machine Learning for Web-Scraping Data Outlier Detection

It is more difficult to detect outliers from web-scraping data than scanner data because the former involves descriptive text information that may cause product misclassification problems. Unlike scanner data with UPC identification, the web-scraping data needs to be cleaned and classified into specific groups before outlier detection can occur. Accordingly, while we can keep similar machine learning procedures as used for scanner data, we need one more layer of product classification using text information. With the web-scraping data on DVDs, we document our methodology with its architecture shown by Figure 5.

Figure 5: Architecture of Outlier Detection for Web-Scraping Data


Text Vectorisation
Method: Oc2Vec Model
Result: We transform qualitative text information on goods attributes to quantitative vectors.
Product Classification
(Text Density Based Clustering)
Method: DBSCAN
Result:. Clusters of goods generated for ordering, with some goods types detected as outliers.
Between Group Ordering
Ordering groups of goods based on ID, date or price
Result: Goodss Groups are ordered and referenceable.
Within Group Ordering:
Ordering goods types within group based on date
Result: Subsets of monthly goods price and observations.
Outlier Detection Round 1
(Within same observation period)
Method: DBSCAN
Result: Outlier detection from observations in each month and generation of monthly price data
Outlier Detection Round 2
(Time series over the whole period)
Method: DBSCAN
Result: Further outlier detection from time series. The price time series is further cleaned from outlier impact.

5.1 Text Vectorisation

We use Doc2Vec and text density-based clustering methods to vectorise the text information on goods characteristics and identify goods groups for further outlier detection analyses.

Based on parameters from the Gensim Python package, we train a Doc2Vec model to derive sentence vectors from descriptive texts of the 120,728 entries on DVDs.

5.2 Product Classification Based on Characteristics Analysis

After converting text information to document vectors, we use the density-based spatial clustering of applications with noise (DBSCAN) algorithm to detect outliers and cluster the data into specific goods.

The steps are as follows: We remove all punctuation marks, set epsilon equal to1 and MinPoint equal to 3 to conduct DBSCAN procedures to identify outliers. There are small amounts of entries without description texts on attributes, we link them with the goods with description texts and cluster them into similar product types. We find that 2-3% of the newly generated good types cannot be classified into any distinctive goods groups and treat them as outliers. We get 99 clusters of goods, each of which contains at least three similar good types. We identify them with ID 1-99 and prepare their price observations for further examination.

5.3 Between Group Ordering

We first access the DVD price data generated in Section 5.2 and construct monthly observations based on the date information. We then order the generated data based on ID, date, or price. The DVDs observations are listed and referenceable over time and across different goods groups. In the following sections, we will use DVD product with ID 2 (Good 2) with observations between October 2018 and April 2019 for illustration purposes.

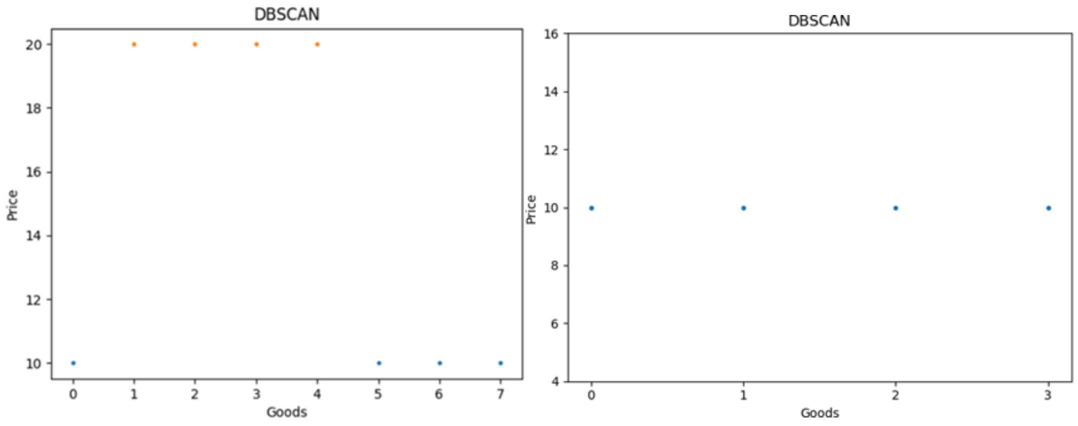
5.4 Within Group Ordering

We order the goods within the same group based on the date information. We can have monthly subsets of DVDs data including all observations occurring in a specific calendar month. For Good 2, we have 5 subsets reflecting the 5 months selling periods in October to December 2018, March 2019 and April 2019.

5.5 First Round Web-Scraping Data Outlier Detection

We keep using DBSCAN algorithm with epsilon set to 1 for outlier detection from price observations within the same months and illustrate the results for October and November 2018 in Figure 6. In most cases, DBSCAN is capable of directly detecting some outliers within the same month. However, when some very high or low prices are clustered in a small group, our DBSCAN algorithm needs to explore information from the previous or the following period for outlier detection. In October 2018 there were 8 similar DVDs with half being sold as £20 and another half as £10. While it is very difficult to identify which price is the outlier based only on the information within the same month, our algorithm can explore information from the following month for reference. Given that in November all 4 DVDs are priced at £10, our algorithm suggests that 4 observations in October priced at £20 are outliers. The £10 change over one month is worthy of further examination as it may be due to half price offers, special edition, or genuine abnormal price level changes that may affect the consumer price index.

Figure 6: Within Group Outlier Detection Results (DVD2, Oct and Nov 2018)

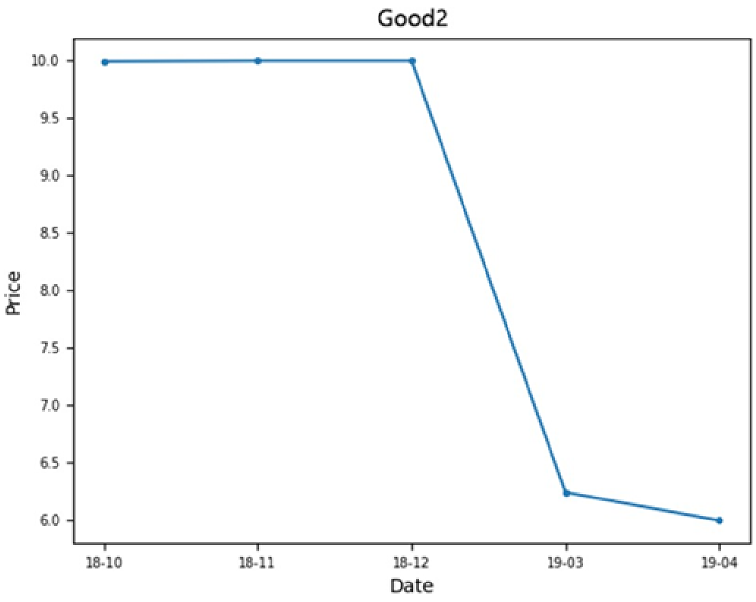


5.6 Second Round Web-Scraping Data Outlier Detection

In this section, we use DBSCAN (epsilon = 1 and MinPoint=3,) for outlier detection from prices of same product over the whole observation periods in 2018 and 2019.

Based on Figure 7, with only several monthly observations to a specific web-scraping product, we cannot detect outliers either from DBSCAN or standard ONS methods (ONS 2019). To increase the number of observation periods, we can either keep collecting data and have a longer observation period or try higher frequency observations, turning 3-7 monthly observations to 13-30 weekly observations like we do with scanner data.

Figure 7: Between Group Outlier Detection Results (DVD2, Oct 2018 – April 2019)



6. Summary and Proposal

This research presents a methodological framework for detecting outliers from alternative price data within the same framework. With examples from both scanner data and web-scraping data, we demonstrate that outliers can be detected with a combination of natural language processing and clustering methods. We can use the methodology to generate improved price indices by providing standard price time series with outliers addressed at the product level.

Looking ahead, although our methodology does not directly address seasonal effects, the issues can be easily tackled by standard time series models when we accumulate enough observation periods. The epsilon value setup of DBSCAN algorithm based on 2.5 times standard deviation is flexible for ONS staff to make sure that we do not delete seasonal effects blindly.

The framework provides flexible outlier detection methods that could benefit from experts' suggestions (the Delphi method) or a per-good threshold recommendation engine for DBSCAN setups. For example, we defined £1 pound price difference in DVDs as the epsilon value to provide a reasonably good starting point to detect the most obvious outliers, but in line with ONS (2019), experts from ONS can propose using a different epsilon value to detect abnormal price level changes that they think that are significant. The outliers we found can be further detected by experts or through a process, such as the Delphi method, for further examination. With training and improvement, we can have improved the algorithm to detect authentic outliers and reduce the chance of false positives.

We also need to use historical standard price indices to compare the potential indices based on alternative data. We could further identify whether we missed some outliers and adjust the algorithm for better outlier detection.

There is room for further development in terms of detecting outliers in price changes. While the current outlier studies focus on price levels instead of price changes, it is possible to conduct data transformation to reflect price changes, define parameters based on price changes, and detect outliers within the same framework.

Longer term, it may be necessary to explain the economic reason behind outliers which requires more data observations to support thorough econometric analysis based on multivariate models.

References

Bojanowski, P., Grave, E, Joulin, A. and T. Mikolov (2016) Enriching Word Vectors with Subword Information, arXiv:1607.04606

Boshoff, J., Mao, X. and G. Young (2020) Outlier Detection Methodologies for Alternative Data Sources: International Review of Current Practices, ESCoE Technical Report 07. Link: https://www.escoe.ac.uk/publications/outlier-detection-methodologies-for-alternative-data-sources-international-review-of-current-practices/

Charlton, L. (2020) Anomaly Detection Literature Review, Working Paper Subject to Request to ONS.

Devlin, J., Chang, M-W., Lee, K., and K. Toutanova (2018) BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding, arXiv:1810.04805v2

Grubbs, F. E. (1969) Procedures for Detecting Outlying Observations in Samples, *Technometrics*,11(1), pp. 1-21.

Joulin, A., Grave, E., Bojanowski, P., and T. Mikolov (2016) Bag of Tricks for Efficient Text Classification, arXiv:1607.01759

Le, Q. V. and T. Mikolov (2014) Distributed Representations of Sentences and Documents, *Proceedings of the 31st International Conference on Machine Learning*, PMLR 32(2), pp. 1188-1196.

Liu, F. T., Ting, K. M. and Z., Zhou, (2008) Isolation Forest, 2008 *Eighth IEEE International Conference on Data Mining*, pp. 413-422.

Maddala, G. S. (1992) *Introduction to Econometrics*, 2nd ed. New York: MacMillan.

Mayhew, M. and G. Clews (2016) *Using Machine Learning Techniques to Clean Web Scraped Price Data via Cluster Analysis*, ONS Survey Methodology Bulletin.

Mikolov, T., Chen, K., Corrado, G., and J. Dean (2013) Efficient Estimation of Word Representations in Vector Space, arXiv:1301.3781

Office for National Statistics (ONS) (2019) Consumer Prices Indices Technical Manual, 2019. Link: https://www.ons.gov.uk/economy/inflationandpriceindices/methodologies/consumerpricesindicestechnicalmanual2019

ONS (2021) Transformation of Consumer Price Statistics: November 2021. Link: https://www.ons.gov.uk/economy/inflationandpriceindices/articles/introducingalternativedatasourcesintoconsumerpricestatistics/november2021