A data-driven supply-side approach for estimating cross-border Internet purchases within the European Union

Meertens, Q.A.; Diks, C.G.H.; van den Herik, H.J.; Takes, F.W.

DOI
10.1111/rssa.12487

Publication date
2020

Document Version
Final published version

Published in
Journal of the Royal Statistical Society. Series A (Statistics in Society)

License
CC BY-NC

Link to publication

Citation for published version (APA):
Meertens, Q. A., Diks, C. G. H., van den Herik, H. J., & Takes, F. W. (2020). A data-driven supply-side approach for estimating cross-border Internet purchases within the European Union. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, *183*(1), 61-90. https://doi.org/10.1111/rssa.12487

ROYAL STATISTICAL SOCIETY
DATA | EVIDENCE | DECISIONS

Journal of the Royal Statistical Society
Statistics in Society
Series A

J. R. Statist. Soc. A (2020)
183, *Part* 1, *pp.* 61–90

A data-driven supply-side approach for estimating cross-border Internet purchases within the European Union

Q. A. Meertens,

University of Amsterdam, Leiden University, and Statistics Netherlands, The Hague, The Netherlands

C. G. H. Diks,

University of Amsterdam and Tinbergen Institute, Amsterdam, The Netherlands

H. J. van den Herik

Leiden University, The Netherlands

and F. W. Takes

University of Amsterdam and Leiden University, The Netherlands

[Received May 2018. Revised May 2019]

Summary. The digital economy is a highly relevant item on the European Union's policy agenda. We focus on cross-border Internet purchases, as part of the digital economy, the total value of which cannot be accurately estimated by using existing consumer survey approaches. In fact, they lead to a serious underestimation. To obtain an accurate estimate, we propose a three-step data-driven approach based on supply-side data. For the first step, we develop a data-driven generic method for firm level probabilistic record linkage of tax data and business registers. In the second step, we use machine learning to identify webshops based on website data. Then, in the third step, we implement recently developed bias correction techniques that have hitherto been overlooked by the machine learning community. Subsequently, we claim that our three-step approach can be applied to any European Union member state, leading to more accurate estimates of cross-border Internet purchases than those obtained by currently existing approaches. To justify the claim, we apply our approach to the Netherlands for the year 2016 and find an estimate that is six times as high as current estimates, having a standard deviation of 8%. Hence, we may conclude that our new approach deserves more investigation and applications.

Keywords: Cross-border e-commerce; Digital economy; Machine learning; Official descriptive statistics; On-line consumption; Probabilistic record linkage

1. Introduction

The accurate estimation of cross-border on-line consumption has recently become more important for two reasons. First, consumption through on-line channels of both goods and services is increasing within the European Union (EU), especially across borders. More consumers have

Address for correspondence: Q. A. Meertens, Faculteit Economie en Bedrijfskunde, Universiteit van Amsterdam, Roetersstraat 11, Amsterdam 1018 WB, The Netherlands.
E-mail: q.a.meertens@uva.nl

access to the Internet, shipping costs are decreasing and payment services are converging across countries (Marcus and Petropoulos, 2016; Cardona and Duch-Brown, 2016; Martikainen *et al.*, 2015). Accurate estimates of cross-border on-line consumption are of increasing importance for adequately reporting on national accounts by national statistical institutes. Second, cross-border on-line trade is nowadays a highly relevant item on the EU digital single-market policy agenda (European Commission, 2010). Therefore, getting a grip on cross-border on-line consumption through reliable estimates is essential for quantifying the effect of new policies. The need for accurate estimates of indicators of the digital economy within the EU was also emphasized by the European Commision (2015).

1.1. Existing survey-based approaches

Existing approaches for estimating consumption are based on either consumer surveys or business surveys. One EU-wide consumer survey on cross-border on-line consumption is conducted by Ecommerce Europe (`https://www.ecommerce-europe.eu`). In the Netherlands, this survey is conducted by market research institute the Gesellschaft für Konsumforschung (GfK) (`https://www.gfk.com`). It is commissioned by the national e-commerce association in the Netherlands, Thuiswinkel.org (`https://www.thuiswinkel.org`), on behalf of Ecommerce Europe. The estimates of total cross-border on-line consumption are based on asking consumers how much they spent at foreign webshops over a fixed time period in the past.

We argue that such an approach based on consumer surveys will lead to an underestimation of cross-border on-line consumption. We start our argumentation with the observation as made by Gomez-Herrera *et al.* (2014). They showed that one of the main impediments of on-line consumption by consumers within the EU is foreign language, rather than security reasons, shipping costs, geographical distance or available payment services. Consequently, webshops that are selling goods or services in multiple countries typically operate in a country by using a website in the regional language (Schu and Morschett, 2017). Therefore, a consumer cannot distinguish between domestic and foreign webshops, as both will be presented in their regional language.

Hence, we may conclude that a consumer survey approach leads to a downward bias in measuring cross-border on-line consumption. We shall refer to the downward bias of consumer survey approaches as *language bias*. To the best of our knowledge, language bias has only been pointed out before by Minges (2016), who was mainly concerned with the implications for official statistics. Here, we stress the scientific implication: current studies on cross-border on-line consumption and trade within the EU, mostly based on consumer survey data, might draw biased conclusions. We suggest that future studies on cross-border on-line consumption and trade should not be based on data obtained (solely) from consumer surveys. To support that suggestion, the goal of this paper is to construct a new and reliable methodology to obtain more accurate estimates of cross-border on-line consumption within the EU. Here, we initially focus on the consumption of goods (Fig. 1).

The first reaction to addressing language bias is to use business surveys instead of consumer surveys (Minges, 2016). However, we believe that this would be unsatisfactory, for two reasons. First, measuring cross-border on-line consumption of consumers in a single country by business surveys requires large companies within the EU to report their sales to consumers per EU member state. This places a large administrative burden on companies. Second, the approach poses significant challenges to any correction for sampling probabilities and biases if, for example, the population of the existing EU-wide information and communication technologies (ICT) in survey enterprises (`https://ec.europa.eu/eurostat/`

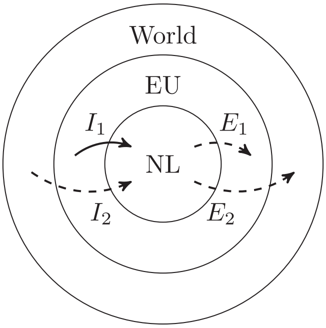

Fig. 1. The four types of cross-border flows of goods crossing an EU member state, e.g. the Netherlands (NL): the paper focuses on flow I_1 (→), the import of goods from other EU member states

cache/metadata/en/isoc_e_esms.htm) would be used. The referenced population is the result of sampling and stratification with respect to economic activity and either relative turnover or number of employees. The stratified sampling probabilities with respect to size in one country must be transformed into that of the total on-line sales in another country. Given large differences between countries in this regard, it seems infeasible to arrive at accurate estimates of cross-border on-line consumption for each EU member state by using business surveys. In summary, both existing official statistical approaches are inadequate in estimating cross-border on-line consumption.

1.2. Our novel approach

The shortcomings of existing approaches that were discussed above are mostly due to the use of inadequate sources of data. Therefore, on the basis of our findings so far, we believe that data used for estimating cross-border on-line consumption should at least meet the following three requirements. First, the data must be based on supply-side information, preventing the aforementioned language bias. Second, the data must be collected for accurately measuring the sales of companies across borders. A reliable administrative or other integral data source would be preferable, since such data prevent having to deal with sampling issues. Third, the data would have to be available to national statistical institutes across the EU, enabling harmonized estimation methods across member states.

Motivated by these three requirements, we propose the use of tax returns filed by foreign companies (subsequently referred to as *tax data*). The EU system of value-added tax (VAT) states that any company that is both established in the EU and involved with cross-border intra-community supplies to consumers must pay VAT in the country of destination and through filing a tax return (European Commission, Council Directive 2006/112/EC). The threshold value on total turnover from sales to consumers, above which filing a tax return is mandatory, is either €35000 or €100000, depending on the country of destination. For foreign companies selling to consumers in the Netherlands, the threshold value on total sales in the Netherlands equals €100000. The Dutch Tax and Customs Administration collects such tax returns, which are then made available to Statistics Netherlands. Other EU member states will have similar, but not the same, data collection procedures, which we shall not discuss. We emphasize that using tax data restricts us to measuring cross-border *on-line consumption of goods* (henceforth referred to as cross-border Internet purchases).

The main challenge in using tax data is to identify webshops. For this identification, we propose an approach consisting of three steps. In the first step, the aim is to select the companies that

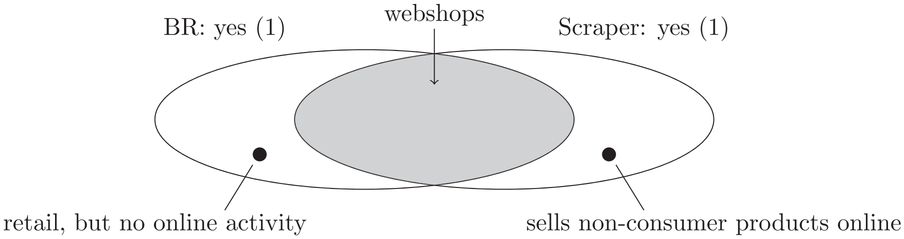
BR: yes (1)
webshops
Scraper: yes (1)
retail, but no online activity
sells non-consumer products online

Fig. 2. Combining the predictions by the business register (BR) with those by websites (scraper)

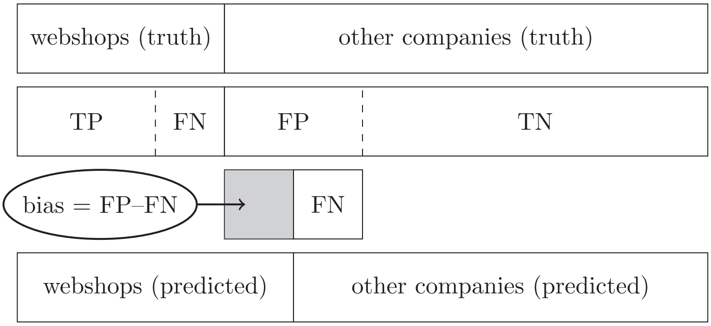

Fig. 3. Initial distribution (top row) of true labels (webshop or other company) and the final distribution (second and bottom row) of predicted labels: the bias in the predicted number of webshops (▨) results from a difference in the number of false positive, FP, and false negative, FN, predictions, showing that the bias is 0 if and only if precision equals recall

are economically active in retail trade, according to the *nomenclature statistique des activités économiques dans la Communauté Européenne* ('NACE') (revision 2). (NACE, revision 2, is the statistical classification of economic activities in the EU; see also `http://ec.europa.eu/eurostat/documents/3859598/5902521/KS-RA-07-015-EN.PDF`.) This is achieved by probabilistic record linkage (at firm level) of the tax data with a business register of retail companies that are active in the EU. In the second step, we use website data (obtained by web scraping) to confirm or complement the result from the first step. We apply machine learning in both of the first two steps to maximize the accuracy of the predictions. The results from the first two steps are combined by taking the intersection of the results (Fig. 2). In the third step, we implement recently developed bias correction techniques (Fig. 3) that have hitherto been overlooked by the machine learning community. Below, we discuss our methodological contributions to the extant literature in each of the three steps, leading to our main contribution.

1.3. *Related work and methodological contributions*

Here, our methodological contributions to the scientific literature on probabilistic record linkage, web scraping and machine learning are discussed. The main scientific contribution is the contribution of the three methods and their incorporation in official statistics, which we point out subsequently.

1.3.1. Probabilistic record linkage

Firm level record linkage in the absence of unique identifiers occurs often in economic research. One of the first well-known examples is the National Bureau of Economic Research's patents data project (Hall *et al.*, 2001). There, the matching was done mostly manually, which was 'one of the most difficult and time-consuming tasks of the entire data construction project'. Since that study, many automated alternatives, using approximate string matching algorithms, have been suggested. Recent examples include Bena *et al.* (2017) and Tarasconi and Menon (2017). The general term for these approximate methods is *probabilistic record linkage*, which was first proposed in the seminal work by Fellegi and Sunter (1969). Now, two issues arise in applications of approximate string matching algorithms:

(a) how to choose a similarity measure and
(b) how to choose an optimal similarity threshold.

The second issue is an optimization problem, that can be solved by using machine learning. Balsmeier *et al.* (2018) proposed to use k-means clustering for this. We shall improve on this work by comparing a wider range of machine learning algorithms and selecting the algorithm that best fits the data. To overcome the second issue we suggest combining the results of multiple similarity measures. These results can be used as features in the machine learning algorithm that optimizes the similarity threshold to choose.

An interesting alternative is to use data from the Internet, as suggested by Autor *et al.* (2017). In brief, their approach entails finding the Uniform Resource Locator (URL) of a company's website by entering the company name into the Internet search engine Bing.com and then using the URL as a unique identifier in the matching process. The advantage of such an approach is that it does not use string matching algorithms at all but instead assumes that the variations in spelling of company names are stored in the database of the Internet search engine. Hence, it solves the two issues of using string matching algorithms at once. However, there are at least two disadvantages of the approach. First, the result of entering a legal company name in an Internet search engine might not always imply that the company's website is included in the top results, in particular for smaller companies. A second disadvantage is that the results are more difficult to reproduce, as the Internet is a dynamic source of data. We therefore propose a combination of Internet search (in our second step) and string matching techniques (in the first step), so that we can benefit from the advantages of both approaches.

1.3.2. Web scraping

In contrast with our fully data-driven work on firm level record linkage, our work on web-based e-commerce detection uses manually selected features based on expert knowledge, because it is easier to understand and implement. Moreover, we show that the accuracy of our approach is high, ruling out the need to implement highly advanced, data-driven web-based e-commerce detection algorithms that are currently cutting edge (Blazquez *et al.*, 2018). We do use a (pre-trained) machine learning model to find the website of a company on the basis of the legal company name. In this respect, we improve on Autor *et al.* (2017). In addition, similarly to the first step, we compare the goodness of fit of a wide range of machine learning algorithms that use (knowledge-based) features obtained by web scraping from company websites to predict whether a company is a webshop or not.

1.3.3. Machine learning

The first two steps provide an accurate (binary) prediction of whether the company is a webshop or not for each company in the set of tax returns (see Fig. 2). What remains is to aggregate the sales of goods of the identified webshops to obtain an estimate of cross-border Internet purchases by Dutch consumers within the EU. However, this estimate will be biased in general, as Fig. 3 illustrates. The fact that classification-based aggregates are biased is relatively understudied in machine learning. To put it more strongly, we believe that we are the first to voice this observation. A related (and mathematically equivalent) problem has been studied before. For example, in epidemiology (Lash *et al.*, 2009) and land cover mapping (Löw *et al.*, 2015), the effect of classification errors on aggregate estimates has been extensively studied for over at least three decades. To the best of our knowledge, the only work that *generally* discusses this effect is Van Delden *et al.* (2016), admittedly in the field of official statistics (and not in the field of machine learning). In all fields, the same equation for the bias of classification-based aggregates has been derived. Our contribution to machine learning is that we show that the fundamental work by Van Delden *et al.* (2016) can be applied to automated classification algorithms in machine learning as well, leading to far more accurate estimates in general.

1.3.4. Official statistics

The main contribution of our paper is to propose a novel methodology to estimate cross-border Internet purchases within the EU, exploiting data and methods that have hitherto not been used for this. We demonstrate that there is convincing evidence from the Netherlands to show that our methodology results in more accurate estimates than approaches based on consumer surveys. The implementation of our methodology in the entire EU could lead to harmonized and accurate estimates of cross-border Internet purchases, ultimately providing policy makers with more reliable information regarding the EU digital single-market policy agenda.

The remainder of this paper is organized as follows. In Section 2, we describe the data that were used. We also describe how we obtained the test data sets that were used to train the machine learning algorithms. In Section 3, the data-driven methods to identify foreign webshops are discussed. In Section 4, we present the results of applying the approach to the Netherlands and we compare them with results from an existing consumer-based approach, demonstrating the severity of the language bias. Section 5 concludes by discussing implementations for other EU member states and possible future research directions.

The programs that were used to analysis the data can be obtained from

`https://rss.onlinelibrary.wiley.com/hub/journal/1467985x/series-a-datasets`

2. Data

In this section we first describe the supply-side data sets (tax data, websites and the business register) that we used. Then, we show how the training and validation data set were obtained. Finally, we discuss the test data set.

2.1. Supply-side data

Below we discuss three types of supply-side data, i.e. tax data, data from websites and data from the business register.

2.1.1. Tax data

The data that were used to measure cross-border Internet purchases are tax returns filed in the Netherlands by foreign companies that are established in the EU. These tax data contain legal company names and the annual turnover from sales in the Netherlands of goods taxed at low or high tariff (i.e. sales to consumers). The data are extracted from tax returns filed for 2014, 2015 and 2016. The data set contains 197424 filed tax returns from 22440 unique

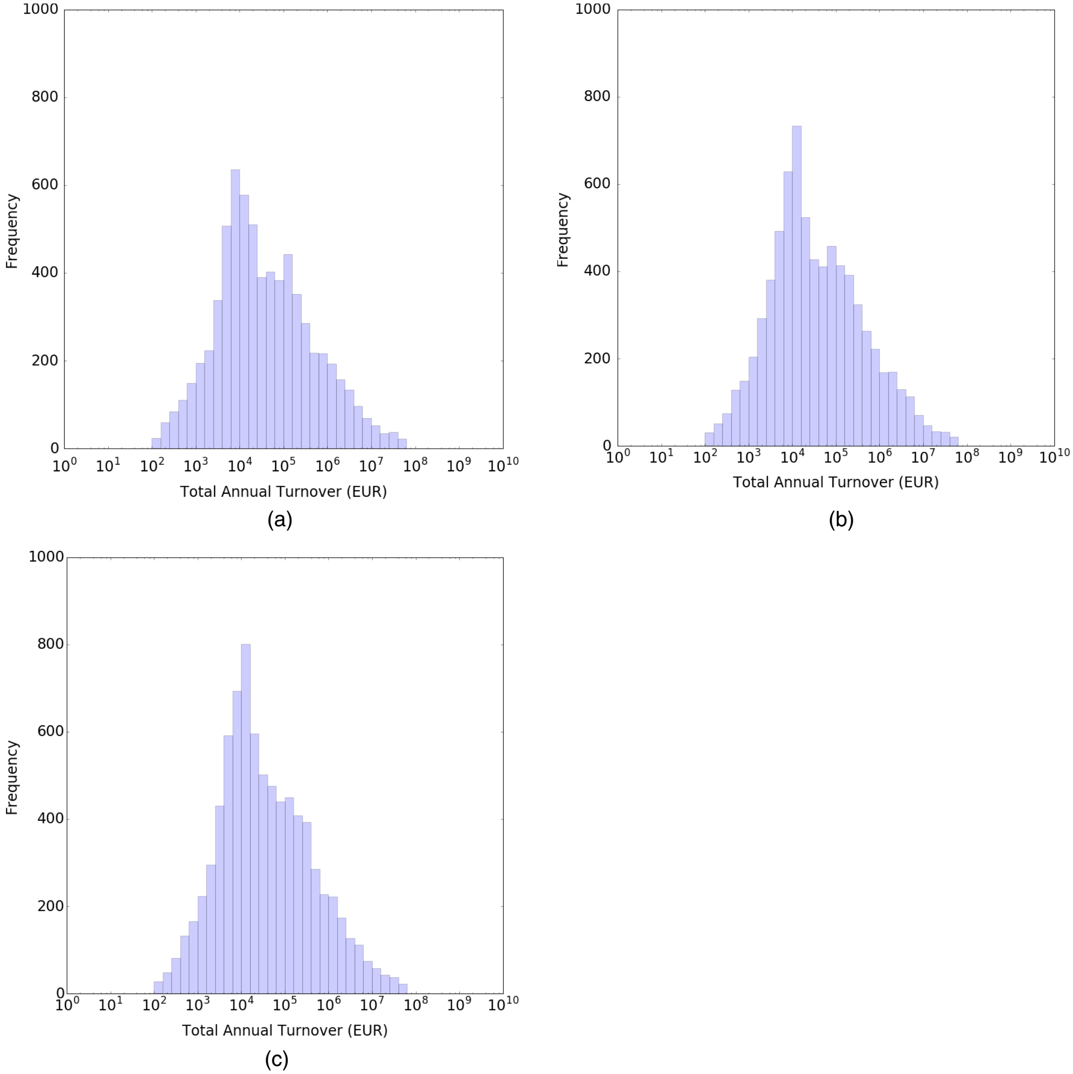

Fig. 4. Distributions of annual turnover of foreign companies that filed a tax return in the Netherlands for (a) 2014, (b) 2015 and (c) 2016: the horizontal axis has a logarithmic scale; companies that reported negative or zero turnover are not shown; because of privacy legislation, bins containing fewer than 20 companies have been removed

companies. These tax data from the Netherlands are not openly available, because of strict privacy legislation. Under severe restrictions (among others, anonymizing) and obeying serious impositions (such as suppressing extreme values), we are permitted to present aggregated figures on the data. When relevant, we reveal the criterion for which we suppressed information. In Fig. 4 we show the distribution of the annual turnover for each of the years 2014, 2015 and 2016. Furthermore, Table 1 displays summary statistics of the tax data from the Netherlands. As the global (cross-border) e-commerce market is rather complicated, we start by making three observations to clarify which flows can and cannot be measured by using the data set of tax returns.

First, smaller sellers might remain unobserved because of the threshold on annual cross-

Table 1. Summary statistics (mean, median and 10th and 90th percentile of annual turnover) of the tax returns filed in the Netherlands by foreign companies in 2014, 2015 or 2016†

Year	$\lvert I\rvert$	$\lvert I_0\rvert$	$\lvert I_{<0}\rvert$	$\lvert I_{>0}\rvert$	*Mean*	*Median*	*10th percentile*	*90th percentile*
2014	16023	8969	86	6968	1904860	25987	1755	1365217
2015	17313	9771	80	7462	1603908	25166	1783	1218926
2016	18939	10626	104	8209	1488351	24790	1879	1143156

†The set of companies filing a tax return in a certain year is denoted by I. The subscript denotes whether the annual turnover in the given period is equal to 0, negative or positive. The number of elements in a set X is denoted by $|X|$.

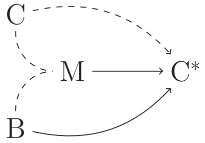

Fig. 5. Possible sales flows if a consumer buys goods on line, showing which flows can (→) and cannot (– – →) be observed by using tax data (C, consumer (seller); B, business (seller); M, marketplace (seller); C*, consumer (buyer)): when the sales are facilitated by an on-line marketplace (e.g. Amazon), the distinction between sales by businesses and sales by consumers cannot be made; therefore, the estimate of cross-border Internet purchases based on tax data might contain transactions from consumers to consumers

border on-line sales to Dutch consumers of €100000. This is particularly problematic as many such small sellers exist: a fact referred to as the long tail of electronic commerce (Bailey *et al.*, 2008; Oestreicher-Singer and Sundararajan, 2012). Many such small sellers use marketplaces (e.g. Amazon) as intermediaries. Now, the distance sales of such marketplaces typically exceed the annual turnover threshold (of €100000). Therefore, they must file tax returns and they show up in the data. Fig. 5 shows in more detail which sales can and cannot be observed by using tax data as the primary source of data to estimate cross-border Internet purchases.

Second, one might wonder to what extent Internet purchases for foreign companies that also have brick-and-mortar stores in the Netherlands show up in the data. Many such multinational companies exist, but most of them have organized their Internet sales from the Netherlands, as a result of which the monetary transaction from consumer to company does not cross borders. In some cases, the Internet sales are organized from outside the Netherlands. The consumer pays a foreign business entity and therefore the sales show up in the data, provided that the restrictions of threshold (€100000 annual turnover) and location (established within the EU) are met. Hence, the trade flows that we observe coincide with the definition of import in the national accounts.

Third, it should be mentioned that our approach can measure cross-border Internet purchases only of companies established within the EU (see Fig. 1). Therefore, purchases at webshops in, for example, China are not included in our estimates. This limitation is noteworthy, as the global e-commerce exports from China are growing vastly nowadays (Ma *et al.*, 2018).

2.1.2. Websites

The website of a company should be a clear indication of whether the company is a webshop or not. We shall use machine learning to distinguish websites of webshops from other websites. The hyper-text mark-up language (HTML) code of the home pages of the websites of companies are the data that we use as input. To obtain these data, we first select the legal company names of the 22400 foreign companies that filed at least one tax return in the Netherlands for 2014, 2015 or 2016. Then, we use *URL retrieval* (Ten Bosch and Windmeijer, 2018) to find the home page of the websites belonging to these companies. Finally, we download the HTML code of the home pages from the Internet. The data were downloaded from the Internet on April 19th, 2017.

2.1.3. *Business register*

We used ORBIS as the business register, which is a global corporate database maintained by Bureau van Dijk (`http://bvdinfo.com/orbis`) and contains detailed corporate information on over 200 million private companies world wide. The database has been claimed to 'suffer from some structural biases' (Ribeiro *et al.*, 2010). However, regarding European companies with an annual turnover of more than €100000, the data set is practically complete (Garcia-Bernardo and Takes, 2018). Data from business registers regarding smaller foreign companies are not needed in our analysis, as these companies do not have to file tax returns in the Netherlands. The ORBIS database is used, because it contains the principal and secondary NACE (revision 2) codes for companies that are established in the EU. The NACE code can be used to select all active (and inactive) companies that are established in the EU and that are principally or secondarily economically active in retail trade. The result is a data set of 6996468 companies, from which companies established in the Netherlands have been excluded. This data set, including each company's country of establishment, was extracted from ORBIS on June 24th, 2017.

For our purposes, any business register containing the company names and country of establishment of every retail company in the EU would suffice, as long as the retail companies (according to NACE, revision 2) can be identified as such. Therefore, we shall henceforth refer to the ORBIS data as *the business register*.

2.2. *Training and validation data set*

To train classification algorithms, a labelled data set is required. Since no such data set existed, we manually constructed it as follows. The tax data contain a classification of economic activity according to the (outdated) Dutch statistical classification of industries from 1974. At first glance, most webshops seemed to be classified as *retail trade*, many as *wholesale trade* and some as another type of industry according to the outdated classification. We constructed a training data set of 180 companies by manually categorizing all companies of which the total annual turnover exceeded an industry-dependent threshold value in at least one year (see the second column of Table 2). The last column of Table 2 displays the number of companies manually categorized per type of industry. In fact, two manual categorizations (see Fig. 2) were made for each company that was included in the training data set presented in Table 2. The first categorization is whether the company is economically active as a retail company according to the business register. We remark that the economic activity reported in the business register might be different from that found in the tax return data set. The second categorization is whether the company is actually a webshop or not, based on manually searching the Internet.

Within the training data set, 76 webshops were identified. Their total turnover in 2016 was equal to €724542550. The validation data set is obtained from the training data set by applying stratified fivefold cross-validation. This is described in more detail in Section 3.1.

2.3. *Test data set*

To assess the goodness of fit of a classification algorithm, we constructed a labelled *test data*

Table 2. Number of companies per industry class (using the Dutch statistical classification of industries (from 1974)) included in the training data set†

Industry (1974)	*Threshold (€)*	*Count*
Retail trade	1 million	100
Wholesale trade	20 million	30
Other	50 million	50
Total	—	180

†The threshold value of the annual turnover is used as a selection criterion for a company to be included in the training data set.

Table 3. Number of companies per industry class (using the Dutch statistical classification of industries (from 1974)) included in the test data set†

Industry (1974)	*Total frequency*	*Count*	*Webshop count*
Retail trade	1393	19	6
Wholesale trade	3329	20	1
Other	17718	40	6
Total	22440	79	13

†The frequency of each industry class in the tax data is included, as well as the number of identified webshops per industry class in the test data set.

set as follows. For *retail trade* and *wholesale trade*, 20 companies that were not in the training data set were randomly selected. (One duplicate retail company had to be removed.) For *other*, 40 companies were selected. The companies selected have been manually categorized, following the same approach as discussed for the companies in the training set. The results of the manual categorization are shown in Table 3.

3. Methods

In this section we discuss the data-driven methods that were used to estimate cross-border Internet purchases within the EU by Dutch consumers. The methods, which can be applied to any other EU member state as well, are presented in three parts. In Section 3.1, the methods that were used to estimate the industry class for companies in the data set of tax returns are specified. Section 3.2 outlines how we accurately estimate webshop turnover. We do this by correcting for biases introduced by inaccuracies in the methods that were used to identify foreign webshops. In Section 3.3, we summarize our data-driven supply-side approach for measuring cross-border Internet purchases within the EU.

3.1. Estimating the industry class

The general set-up is as follows. Consider a population of n companies indexed by a set I. For a company $i \in I$, the industry class is denoted by $s_i \in H$. The set H consists of only two industry classes, namely webshops ($s_i = 1$) and other companies ($s_i = 0$). The total turnover from sales of goods as reported in the tax returns in year t by company i is denoted by $y_{i,t}$. In the tax data, $y_{i,t}$ is given for each $i \in I$ and each $t \in \{2014, 2015, 2016\}$. The goal is to estimate the annual turnover from sales of goods of the foreign webshops in the data set for each year t, given by

$$\sum_{i \in I} s_i y_{i,t}. \tag{1}$$

We assume that the classification s_i does not depend on t, although a company's economic activity might in reality change over time, e.g. when two companies merge. However, the specific case of merging companies is handled correctly, as the merged company will show up as a new company $i \in I$ in the data. Other causes for changes in economic activity are not corrected for. This might be refined in future work by determining the company's classification periodically (e.g. once a year).

The challenge of estimating expression (1) is that the industry classes s_i are not observed but must be estimated instead. We propose to estimate s_i in two different ways (in Section 3.1.1 by the business register and in Section 3.1.2 by websites) and combine the two estimates into a final estimate of industry class s_i (see Fig. 2). The combined estimate of s_i is used to evaluate expression (1).

3.1.1. Estimating the industry class by the business register

We assume that the sales of goods as reported in the tax returns filed by foreign companies registered (in the business register that we use) as a retail company according to the NACE (revision 2) code are precisely the cross-border Internet purchases within the EU by the consumers of the EU member state under consideration. In other words, if a company $i \in I$ is registered as a retail company in the business register, we set the estimated industry class by the business register $\hat{s}_i^{\text{BR}}$ to 1. If not, we set $\hat{s}_i^{\text{BR}} = 0$.

The challenge is that we cannot simply look up a company $i \in I$ in the business register, as the tax data and the business register do not share a common unique identifier. The two data sets must be merged by matching legal company names. The following four issues then arise. First, the type of business entity might be registered differently in both data sets (e.g. LTD or LIMITED). Second, the name of a company might be spelled differently in both data sets (e.g. Muller or Mueller) and taking such differences into account (i.e. performing probabilistic record linkage) is computationally expensive. Third, we must choose which string distance metric to use for quantifying such differences numerically. Fourth, a threshold on the permitted number of spelling differences between names belonging to the same company must be determined. To overcome these four issues, we propose the following four-step approach (I–IV), where each step addresses the corresponding issue. Implementation details can be found in Appendix A. We emphasize that only step I is specific to applications of firm level record linkage. For other applications considering probabilistic record linkage, steps II–IV of our approach may directly be used.

3.1.1.1. Step I: stemming company names. First, as a preprocessing step, non-alphanumeric characters are replaced by spaces, all leading, trailing and duplicate spaces are removed, and all characters are converted to lower case. Then, we remove the type of business entity (e.g. LTD) from the legal company name. For this, we may apply suffix stripping (or *stemming*) techniques, because the type of business entity comprises the end of a legal company name. Our data-driven stemming approach is inspired by Lovins (1968) and Porter (1980), where the latter is claimed to be 'the most common algorithm for stemming English' (Manning *et al.* (2009), page 32). Finally, three values are stored for each company in the tax data and the business register. Taking the German company *Muller GmbH*, for example, the following three values are stored: `stem = 'muller'` ; `suffix = 'gmbh'` ; `suffix_class = 'LTD/DE'` . The *suffix class* indicates the type of company (using British equivalents) and the company's EU member state of establishment.

3.1.1.2. Step II: locality-sensitive hashing. The variety of possible spelling differences in names of companies from the entire EU is huge, so manually formulating rules for matching tax data and the business register on company names is infeasible. To automate name-based record linkage we use *approximate string matching*; see, for example, Cohen *et al.* (2003) for an overview. This entails measuring the distance between names (from now on, strings) viewed as elements of a (typically high dimensional) metric space. The approximate string match (according to a metric d) in the business register of a string s from the tax data would be the string t in the business register that minimizes $d(s,t)$ for t in the business register. The problem is that the value $d(s,t)$ must be computed for each pair (s,t), which is computationally expensive. In our case, $22440 \times 6996468 \approx 1.57 \times 10^{11}$ values are computed, which may take up to several days on a regular machine, depending on which string distance metric is used.

An efficient and elegant approach is to use locality-sensitive hashing (LSH) schemes, which can be thought of as randomized dimensionality reduction preserving string distance. We shall use the famous LSH scheme MinHash (Broder, 1997), which is locality sensitive for the Jaccard distance on character n-grams, or n-shingles (Leskovec *et al.* (2014), chapter 3).

Although MinHash enables a faster approximate evaluation of the string distance metric, it does not yet reduce the number of evaluations that are required to match the two sources of data. To achieve this, we apply the LSH forest data structure (Bawa *et al.*, 2005) on the results of MinHash by using the business register. This data structure can then be queried to retrieve, for any string s (from the tax data) and any natural number m, the m approximately most similar strings in the input data set (the business register) according to any metric that induces an LSH family (including MinHash). Choosing $m = 100$, the approximate string matching and LSH techniques have reduced the number of evaluations from 1.57×10^{11} to $22440 \times 100 = 2.24 \times 10^{6}$ (spending only about 80 min of wall clock computation time on a regular machine).

3.1.1.3. Step III: combining string distance metrics. What remains, is to find the closest match in the remaining $m = 100$ companies from the business register for each company from the tax data, according to some string distance metric d. As we are not necessarily interested in the closest match itself, but simply in the binary outcome 'match'–'no match', we store only the minimum distance. For other applications where the match itself is of interest (e.g. in general record linkage problems), store the m-vector of distances and apply the remainder of our approach accordingly.

Two choices must be made in advance. First, some string distance metric must be selected. Second, some threshold value for $d(s,t)$ must be determined, above which we consider the approximate match a real match. In existing work on probabilistic record linkage of firm level sources of data, these two choices are made manually, and typically somewhat arbitrarily. Therefore, the accuracy of the results will not be as high as possible, in general. To increase the accuracy, we propose the following general *data-driven* approach: consider multiple string distance metrics at once and let a machine learning algorithm determine the optimal threshold values. Details on which string distance metrics we have combined can be found in Appendix A.

3.1.1.4. Step IV: machine learning. As mentioned in step III, we propose to use machine learning to find the optimal threshold values for string distance metrics (above which we consider an approximate match a real match). The aim is to find a classification algorithm $\hat{s}_i^{\mathrm{BR}}$ that accurately predicts the industry class $s_i^{\mathrm{BR}} \in H$ (i.e. whether company $i \in I$ is registered in the business register as a retail company). The algorithm will use the eight-dimensional vectors of distances that are constructed in step III as the features (see Appendix A). Recall that, for each company in the training set and the test set, the class s_i^{BR} was observed by manually searching the business register.

We propose the following data-driven approach to select a classification algorithm and corresponding algorithm parameter settings that are optimal in predicting s_i^{BR}. First, 10 of the most commonly used classification algorithms are selected to be examined. We note that this selection is not exhaustive and it might be extended in future work. The 10 classification algorithms that we consider are listed in Table 4. We consider the linear classification algorithms logistic regression, LR, linear discriminant analysis, LDA, and linear support vector classification, LinSVC. The non-linear algorithms that were implemented are k nearest neighbours, kNN, multinomial naive Bayes, MNB, quadratic discriminant analysis, QDA, and support vector classification with radial basis function kernel, RBFSVC. Furthermore, we examine three ensemble algorithms, namely random forests, RF, gradient boosting, GB, and AdaBoost, AB. The details of the specifications of the classification algorithms can be found in, for example, Witten *et al.* (2017), Han *et al.* (2011) or Hastie *et al.* (2009). We used the Python library scikit-learn (`http://scikit-learn.org/`, version 0.19.1) to implement the 10 classification algorithms.

For each of the 10 algorithms, a grid of parameter combinations to be examined was specified. These grids are depicted in Table 5. For precise parameter specifications we refer to the scikit-learn documentation.

For each algorithm and each parameter combination in the parameter grid, stratified fivefold cross-validation is performed on the training data set. Cross-validation is used to prevent overfitting. The choice of using five folds is based on Breiman and Spector (1992). It might introduce more variance than choosing 10 or 20 folds (Kohavi, 1995). However, because of the small size of the training data set, choosing 10 or 20 folds might lead to unstable results. Therefore, we have chosen to use *stratified* fivefold cross-validation to reduce the variance, as suggested by Kohavi (1995). Furthermore, we optimize parameter settings by using mean F1-scores over the five folds. We prefer F1 over accuracy because of the low base rate of webshops in the entire data set. We do not use the common metric AUROC to optimize param-

Table 4. Overview of the 10 classification algorithms that we consider, including whether we refer to it as a linear, non-linear or ensemble algorithm

Type	*Algorithm*
Linear	Logistic regression, LR
	Linear discriminant analysis, LDA
	Linear support vector classification, LinSVC
Non-linear	k nearest neighbours, kNN
	Multinomial naive Bayes, MNB
	Quadratic discriminant analysis, QDA
	Support vector classification with radial basis function kernel, RBFSVC
Ensemble	Random forests, RF
	Gradient boosting, GB
	AdaBoost, AB

Table 5. Overview of the parameter grids for the algorithms examined†

Algorithm	*Parameter grid*
LR	Penalty, {l1, l2}; C, $\{0.001, 0.01, 0.1, 1, 10\}$
LDA	Non-parametric
LinSVC	C, $\{0.001, 0.01, 0.1, 1, 10\}$
kNN	k, $\{1, 3, 5, \ldots, 39\}$
MNB	α, $\{10^{-10}, 0.01, 0.1, 1\}$
QDA	Non-parametric
RBFSVC	C, $\{0.01, 0.1, 1, 10, 100\}$; γ, $\{0.001, 0.01, 0.1, 1\}$
RF	n, $\{50, 100, 200, 500\}$; d, $\{1, 2, 3, \ldots, 8\}$
GB	n, $\{50, 100, 200, 500\}$; d, $\{1, 2, 3, \ldots, 8\}$; λ, $\{0.01, 0.1, 1\}$
AB	n, $\{50, 100, 200, 500\}$, d, $\{1, 2, 3, \ldots, 8\}$; λ, $\{0.01, 0.1, 1\}$

†In estimating the algorithms LR, LinSVC, RBFSVC, RF and AB, the two-class weighting schemes uniform and balanced were also included in the parameter grid. See the scikit-learn documentation (`http://scikit-learn.org/`, version 019.1) for parameter specifications.

eter settings, as it is known possibly to mask poor performance when facing imbalanced data (Jeni *et al.*, 2013). As our data are in fact strongly imbalanced, because of the low base rate of webshops, it does not seem wise to use AUROC as the optimizing metric. Moreover, optimizing AUROC does not, in general, imply optimizing F1 (Davis and Goadrich, 2006). Thus, for each algorithm the parameter setting that maximizes the mean F1-score over the five folds is selected. Subsequently, the mean and standard deviation of F1-scores over the five folds between the 10 optimal classification algorithms are compared.

Finally, both the mean F1-score and the standard deviation of F1-scores over the five folds are considered in choosing the final classification algorithm and corresponding parameter settings. If necessary, the local behaviour on the parameter grid is examined to reduce the standard deviation of F1-scores over the five folds. This final classification algorithm is then trained on the entire training data set. The trained classification algorithm is used to compute the estimate $\hat{s}_i^{\mathrm{BR}}$ for each company i that is not included in the training data set. Recall that $\hat{s}_i^{\mathrm{BR}}$ is an estimate of s_i^{BR}, which indicates whether company i is registered as a retail company in the business register. In practice, it might be different from the true industry class s_i that we aim to estimate. For this reason, we propose to estimate the industry class by websites as well, resulting in a second estimate $\hat{s}_i^{\mathrm{W}}$.

3.1.2. *Estimating the industry class by websites*

We assume that a webshop can be identified by a shopping cart on the home page, referred to as such in the underlying HTML code. If a shopping cart is found on the website of company $i \in I$, we set the estimated industry class by websites $\hat{s}_i^{\mathrm{W}}$ to 1. If not, we set $\hat{s}_i^{\mathrm{W}} = 0$. For this classification, we propose the following three-step approach. First, as the tax data do not contain the URL of the website of a company, we implement a method for finding this URL based on the legal company name. Second, web scraping is used to look for a shopping cart on the website. Third, as the first two steps are not flawless, the machine learning approach from Section 3.1.1.4 is used to minimize errors. The following three parts describe these three steps in more detail.

3.1.2.1. Step I: finding a company's website. In tax data from the Netherlands, a URL of the home page of the company is not available. Therefore, Statistics Netherlands has developed

URL retrieval software to retrieve the URL of the home page of a company on the basis of the legal company name (Ten Bosch and Windmeijer, 2018). The legal company name is first processed by Google's search application programming interface, returning a list of several URLs, each equipped with a title and a description. The URLs are then ranked according to a *matching score* between 0 (definitely not a match) and 1 (definitely a match), which is computed by a random-forest algorithm. The algorithm is trained by using a set containing 1000 Dutch company names (of companies from various industries and varying in size, i.e. the number of employees) and the URL of their website. We emphasize that the Dutch language of the training set is not necessarily an issue, as most foreign webshops selling to Dutch consumers will have a Dutch version of the website (see Section 1). For each company, the URL with the highest assigned matching score is returned and the corresponding matching score is stored.

3.1.2.2. Step II: searching for a shopping cart. For each company, the code of the URL that is found in step I is downloaded as a raw text file. In the raw text file, the occurrences of variations of the words *shop* and *cart* in Dutch, English and German are counted. The full list is *winkel*, *wagen*, *mand*, *shop*, *cart*, *bag*, *basket* and *warenkorb*. The choice of these three languages is based on the fact that most Dutch citizens mostly speak only Dutch, English and/or German. Note that, in modern information retrieval, it is more common to count the occurrences of all words found in a document (see Manning *et al.* (2009), chapter 6). We have chosen not to follow this approach, as it would lead to serious dimensionality issues; the number of different terms (words) would be much larger than the number of documents (websites of companies in the training set).

3.1.2.3. Step III: machine learning. The final step is to find a classification algorithm $\hat{s}_i^{\mathrm{W}}$ that can accurately predict the industry class $s_i \in H$. The true, unobserved industry class $s_i^{\mathrm{W}} \in H$ represents whether company $i \in I$ is a webshop. Recall that, for each company in both the training and the test set, the class s_i^{W} was observed by manually searching the Internet.

Before training a classification algorithm, the counts of the words are transformed to real numbers in the interval [0,1] by using (normalized) term frequency × inverse document frequency (TF × IDF) (see Witten *et al.* (2017), page 314, for a definition). To prevent division by 0 in computing IDF, a single document containing each of the eight words once is added to the data. The eight TF × IDF values and the maximum matching score are used as features in fitting classification algorithms on the training data. The machine learning approach is identical to that described in Section 3.1.1.4.

3.1.3. Constructing the final estimate of the industry class

The two selected classification algorithms, each with the optimal parameter setting, are trained on the entire training data set. The trained models are used to compute $\hat{s}_i^{\mathrm{BR}}$ and $\hat{s}_i^{\mathrm{W}}$ on the remaining part of the data set. Companies whose features, which are needed for one of the two algorithms, are (partially) missing receive the value −1 as prediction, to be interpreted as 'missing'. It happens for $\hat{s}_i^{\mathrm{BR}}$ if the tax stem of the company has less than three characters. It happens for $\hat{s}_i^{\mathrm{W}}$ if the maximum matching score is below 0.5 or no HTML code was downloaded. The final single categorization $\hat{s}_i$ is obtained by combining $\hat{s}_i^{\mathrm{BR}}$ and $\hat{s}_i^{\mathrm{W}}$ as follows:

$$\hat{s}_i := \begin{cases} -1 & \text{when } \hat{s}_i^{\mathrm{BR}} = \hat{s}_i^{\mathrm{W}} = -1, \\ \hat{s}_i^{\mathrm{BR}} & \text{when } \hat{s}_i^{\mathrm{W}} = -1, \\ \hat{s}_i^{\mathrm{W}} & \text{when } \hat{s}_i^{\mathrm{BR}} = -1, \\ \hat{s}_i^{\mathrm{BR}} \wedge \hat{s}_i^{\mathrm{W}} & \text{otherwise.} \end{cases}$$

The AND operator '$\wedge$' is computed as the minimum of the two integers. It implies that $\hat{s}_i$ categorizes a company as a webshop if and only if the company is categorized as such by both $\hat{s}_i^{\mathrm{BR}}$ and $\hat{s}_i^{\mathrm{W}}$ (see Fig. 2).

3.2. Accurately estimating webshop turnover

Estimating webshop turnover once $\hat{s}_i$ has been estimated seems straightforward: simply use it instead of s_i to evaluate expression (1). However, this straightforward evaluation will result in a biased estimate of webshop turnover. This section aims to estimate and correct that bias, yielding a more accurate estimate of webshop turnover.

3.2.1. *Biased estimation of webshop turnover*

We begin by isolating the bias in estimating expression (1). For companies in the training set, the manual categorization s_i is the true class of company $i \in I$. Hence, rewriting expression (1), the total (annual) cross-border Internet purchases could thus be estimated as

$$\sum_{i \in I_{\mathrm{M}}} s_i y_i + \sum_{i \in I \setminus I_{\mathrm{M}}} \hat{s}_i y_i, \tag{2}$$

where $I_{\mathrm{M}} \subset I$ is the training set of manually categorized companies. The first term is the total turnover of observed webshops in the training data set and the second term is the total turnover of predicted webshops in the rest of the data set.

Now, Fig. 3 illustrates that expression (2) yields (because of the second term) a biased estimate of expression (1). In fact, any aggregate based on the results of a classification algorithm will be a biased estimate of the true value. For a binary classifier, the only exception is when the number of false positive predictions is equal to the number of false negative predictions, which is equivalent to precision and recall being equal. To the best of our knowledge, we are the first to note this in the setting of machine learning.

Before estimating and correcting the bias we introduce the vector notation from Van Delden *et al.* (2016). We write $\mathbf{a}_i$ for the 2-vector $(s_i, 1 - s_i)^{\mathrm{T}}$ and consider the aggregate turnover vector $\mathbf{y} = \Sigma_{i \in I} \mathbf{a}_i y_i$. Similarly, define $\hat{\mathbf{a}}_i$ based on $\hat{s}_i$. Expression (2) will thus become the first component of the estimated 2-vector $\hat{\mathbf{y}}$ given by

$$\hat{\mathbf{y}} := \sum_{i \in I_{\mathrm{M}}} \mathbf{a}_i y_i + \sum_{i \in I \setminus I_{\mathrm{M}}} \hat{\mathbf{a}}_i y_i. \tag{3}$$

In the remainder of this section, only the subset $I \setminus I_{\mathrm{M}} \subset I$ is considered. Hence, any index i will refer to a company that is not in the training set I_{M}. Consequently, the estimate $\hat{\mathbf{y}}$ will be used to refer only to the second term on the right-hand side of equation (3), as the first term does not introduce any bias. Similarly, $\mathbf{y}$ will be used in the remainder of this section to refer to $\Sigma_{i \in I \setminus I_{\mathrm{M}}} \mathbf{a}_i y_i$.

3.2.2. *Classification error model*

To estimate and correct the bias, we follow the approach of Van Delden *et al.* (2016), which did not focus on machine learning algorithms, but it can be directly applied in that setting, as we show below. The approach entails that s_i is considered to be deterministic and $\hat{s}_i$ to be stochastic, conditionally on s_i. They assume the following *classification error model*

$$p_{ghi} := \mathbb{P}(\hat{s}_i = h | s_i = g), \qquad g, h \in H. \tag{4}$$

We emphasize that this assumption is very reasonable if $\hat{s}_i$ is the result of a machine learning algorithm. Such an algorithm is based on assuming a data-generating process, where the independent variable s_i is assumed to be a function of dependent variables (or features) that result in $\hat{s}_i$, plus an error or noise term. This noise term corresponds to the stochastic classification error model above.

In addition, we assume that p_{ghi} does not depend on $i \in I \backslash I_{\mathrm{M}}$. This assumption might be argued to be incorrect for two reasons. First, it is more difficult to find the correct website for a small company than for a large company. Moreover, a small company that is not a webshop might not even have a website. Second, the coverage and quality of the business register for smaller companies is significantly lower than for larger companies. Both reasons imply that the probability of a classification error (more specifically, a false negative classification error) increases as turnover decreases. However, we make the assumption because accurately estimating P for different turnover classes, as suggested by Van Delden *et al.* (2016), requires a far larger training data set than the training data set that we have available.

The resulting 2×2 matrix $P=(p_{gh})_{g,h\in H}$ is estimated as follows. On the test data set, $\hat{s}_i$ is compared with s_i. Denoting by TP, FP, TN and FN the number of true and false positive and true and false negative classifications respectively, the estimator $\hat{P}$ for P takes the form

$$\hat{P}=\begin{pmatrix} \dfrac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} & \dfrac{\mathrm{FN}}{\mathrm{TP}+\mathrm{FN}} \\ \dfrac{\mathrm{FP}}{\mathrm{TN}+\mathrm{FP}} & \dfrac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}} \end{pmatrix}. \tag{5}$$

We next show how to use this estimator to obtain accurate estimates of cross-border Internet purchases.

3.2.3. *Estimating bias and variance*

If we assume the classification error model, it follows that $\mathbb{E}(\hat{\mathbf{a}}_i)=P^{\mathrm{T}}\mathbf{a}_i$ and therefore

$$\mathbb{E}(\hat{\mathbf{y}})=P^{\mathrm{T}}\mathbf{y}. \tag{6}$$

The bias of $\hat{\mathbf{y}}$ as an estimator of $\mathbf{y}$ equals

$$\mathbf{B}(\hat{\mathbf{y}})=\mathbb{E}(\hat{\mathbf{y}})-\mathbf{y}=(P^{\mathrm{T}}-I_2)\mathbf{y}, \tag{7}$$

where I_2 is the 2×2 identity matrix. This shows that, in general, equation (3) yields a biased estimate of $\mathbf{y}$. In fact, the bias is only 0 if either

(a) the classification algorithm does not make any errors (i.e. $P^{\mathrm{T}}=I_2$) or
(b) $\mathbf{y}$ precisely equals an eigenvector of P^{T} corresponding to the eigenvalue 1.

To estimate the bias as given in expression (7), we could use the plug-in estimator

$$\hat{\mathbf{B}}_0=(\hat{P}^{\mathrm{T}}-I_2)\hat{\mathbf{y}}. \tag{8}$$

Following Van Delden *et al.* (2016), we assume that $\mathbb{E}(\hat{P}^{\mathrm{T}})=P^{\mathrm{T}}$ and that $\hat{P}^{\mathrm{T}}$ and $\hat{\mathbf{y}}$ are uncorrelated. It follows that $\mathbb{E}(\hat{\mathbf{B}}_0)=P^{\mathrm{T}}\mathbf{B}(\hat{\mathbf{y}})$; hence the plug-in estimator is a biased estimator of the bias. If we assume that $p_{01}+p_{10}\neq 1$ (and $\hat{p}_{01}+\hat{p}_{10}\neq 1$), then the inverse matrix $Q=(P^{\mathrm{T}})^{-1}$ exists (and $\hat{Q}=(\hat{P}^{\mathrm{T}})^{-1}$ exists). Now, assuming that $\mathbb{E}(\hat{Q})=Q$ and that $\hat{Q}$ and $\hat{y}$ are uncorrelated, an unbiased estimator of the bias is

$$\hat{\mathbf{B}}_1=(I_2-\hat{Q})\hat{\mathbf{y}}. \tag{9}$$

However, correcting $\hat{\mathbf{y}}$ by $\hat{\mathbf{B}}_1$ might increase the variance of (the first component of) the estimator. It might lead to low accuracy in practice. Therefore, Van Delden *et al.* (2016) proposed to find the optimal value $\lambda = \lambda^*$ for which the mean-squared error of the first component of $\hat{\mathbf{B}}_\lambda = (1-\lambda)\hat{\mathbf{B}}_0 + \lambda\hat{\mathbf{B}}_1$ as an estimator of the bias $\mathbf{B}(\hat{\mathbf{y}})$ is minimized for $\lambda \in [0,1]$. For more details on how to derive λ^*, consult Appendix B.

Having found λ^*, we estimate $\mathbf{y}$ (still excluding $\mathbf{y}_\mathrm{M}$) by the first component of the vector

$$\hat{\mathbf{y}}_{\lambda^*} = \hat{\mathbf{y}} - \hat{\mathbf{B}}_{\lambda^*} = \hat{\mathbf{y}} - (I_2 + \lambda^*(\hat{Q} - I_2))(\hat{P}^\mathrm{T} - I_2)\hat{\mathbf{y}} = (2I_2 - \hat{P}^\mathrm{T} - \lambda^*(\hat{Q} - I_2)(\hat{P}^\mathrm{T} - I_2))\hat{\mathbf{y}}. \quad (10)$$

The standard deviation is estimated by the square root of the upper left-hand value in the variance–covariance matrix

$$\hat{V}(\hat{\mathbf{y}}_{\lambda^*}) = (2I_2 - \hat{P}^\mathrm{T} - \lambda^*(\hat{Q} - I_2)(\hat{P}^\mathrm{T} - I_2))\hat{V}(\hat{\mathbf{y}})(2I_2 - \hat{P}^\mathrm{T} - \lambda^*(\hat{Q} - I_2)(\hat{P}^\mathrm{T} - I_2))^\mathrm{T}, \quad (11)$$

Here, the variance $V(\hat{\mathbf{y}})$ of $\hat{\mathbf{y}}$ is estimated by

$$\hat{V}(\hat{\mathbf{y}}) = \mathrm{diag}(\hat{P}^\mathrm{T}\hat{\mathbf{k}}) - \hat{P}^\mathrm{T}\mathrm{diag}(\hat{\mathbf{k}})\hat{P}, \quad (12)$$

where $\hat{\mathbf{k}} = \Sigma_i \hat{\mathbf{a}}_i y_i^2$. The bias of $\hat{V}(\hat{\mathbf{y}})$ as an estimator of $V(\hat{\mathbf{y}})$ is relatively small and therefore is not corrected (Van Delden *et al.* (2015), appendix A4). As the values of $\mathbf{y}_\mathrm{M}$ are not stochastic, expression (11) also yields the standard deviation of the final estimate of $\mathbf{y}$.

3.3. *Summarizing our data-driven supply-side approach*

The proposed data-driven supply-side approach for measuring cross-border Internet purchases within the EU can be summarized as follows. Based on EU VAT legislation, the starting point is a data set of tax returns filed by foreign companies that are established within the EU. These tax data are *supply-side* data as they contain company turnover. Then, the challenge is to identify webshops within the data set of tax returns. We address this challenge in two steps. In the first step, we implement approximate string matching techniques to merge the tax data to a business register of retail companies that are established within the EU. The merging can be viewed as *data-driven* record linkage, as we optimized the performance of the approximate string matching by using machine learning algorithms. In the second step, we use web scraping in combination with machine learning to assess whether a company is a webshop. The outcomes of the two steps are combined to obtain a more accurate estimate of cross-border Internet purchases. Moreover, we use the data to estimate the bias and standard deviation of the estimate. Thus, the data-driven methods applied to the supply-side data yield our data-driven supply-side approach for measuring cross-border Internet purchases within the EU.

4. Results

Below, we present our findings of applying the approach to the Netherlands by estimating cross-border Internet purchases within the EU by Dutch consumers. The section is structured as follows. First, in Section 4.1, the results of training the classification algorithms to estimate the industry class by the business register are presented. Then, in Section 4.2, the same is presented for estimating the industry class by websites. Next, in Section 4.3, we present the results of estimating cross-border Internet purchases by Dutch consumers. It contains the most relevant results of the paper. Finally, in Section 4.4, we compare the results of our data-driven supply-side approach to currently available results from demand-side approaches based on consumer surveys. We interpret and discuss the differences of the resulting estimates of cross-border Internet purchases by Dutch consumers.

4.1. Results from estimating the industry class by the business register

As can be seen from the results in Table 6, machine learning is very well suited for probabilistic record linkage of firm level data. Recall from Section 3.1.1.4 that we compared 10 different machine learning algorithms (Table 4), each evaluated by using multiple parameter settings (Table 5), to estimate the industry class s_i^{BR} by business registers. Table 6 does not include results for MNB; this algorithm assumes discrete features, whereas they are continuous (distances between strings). For each algorithm, we have selected the parameter settings that are optimal in estimating s_i^{BR}, based on the mean F1-score from the stratified fivefold cross-validation. The results in Table 6 show that the mean goodness of fit of the machine learning algorithms are high, with little difference between the algorithms. Moreover, the standard deviations in scores over the folds (which are shown in parentheses) are small.

The final classification algorithm that we use to predict s_i^{BR} is RBFSVC, with parameters $C=100$ and $\gamma=1$ and the balanced class weighting scheme (see Table 6). Observe that this choice not only maximizes the mean F1-score, but also mean precision and mean recall. In particular, the algorithm does not falsely predict positive classifications on the training data set. Moreover, the local behaviour of the mean F1-score of RBFSVC, as a function of the parameters C and γ, is stable around the optimal parameters (see Appendix C).

4.2. Results from estimating the industry class by websites

The results in Table 7 show lower scores and greater difference between algorithms than the results in Table 6. Again, the algorithms (which are now used to estimate the industry class s_i^{W} by websites) are ranked with respect to the optimal mean F1-score over the folds in the stratified fivefold cross-validation. Also, note that the standard deviations of scores over the folds (which are shown in parentheses) are relatively high. Analysing the results more closely, using the (simple) categorization of the machine learning algorithms that were used into linear, non-linear and ensemble algorithms (see Table 4), we make an interesting observation. On the basis of the results in Table 7, all three algorithms from the category of linear methods (LR, LinSVC and LDA) perform less well than the (better performing) algorithms from the other two categories of methods. This suggests that a linear separation of the data points in higher dimensional space does not yield the best classification for unseen data. Therefore, it could be

Table 6. Mean (plus or minus standard deviation) of scores for optimal parameter settings for each of the specified algorithms estimating s_i^{BR}†

Algorithm	*Optimal parameters*	*F1*	*Precision*	*Recall*
RBFSVC	$C=100, \gamma=1$	*0.97 (± 0.03)*	*1.00 (± 0.00)*	*0.94 (± 0.05)*
GB	$n=50, d=1, \lambda=0.01$	0.95 (± 0.02)	0.98 (± 0.03)	0.92 (± 0.03)
kNN	$k=3$	0.95 (± 0.03)	0.98 (± 0.03)	0.93 (± 0.04)
LinSVC	$C=0.01$	0.94 (± 0.02)	0.97 (± 0.03)	0.91 (± 0.03)
LDA		0.94 (± 0.03)	*1.00 (± 0.00)*	0.89 (± 0.05)
LR	$C=1$, L1-penalty	0.94 (± 0.03)	0.97 (± 0.03)	0.92 (± 0.03)
AB	$n=100, d=1, \lambda=0.1$	0.94 (± 0.04)	0.96 (± 0.04)	0.93 (± 0.04)
RF	$n=50, d=4$	0.94 (± 0.04)	0.95 (± 0.04)	0.93 (± 0.04)
QDA		0.93 (± 0.02)	*1.00 (± 0.00)*	0.87 (± 0.03)

†The scoring function F1 (used to rank the results) is used to optimize across the parameter settings in the parameter grid. Each parameter setting is evaluated by using stratified fivefold cross-validation. In each column, the maximum score is highlighted. In the fourth column three scores are the maximum score (rows RBFSVC, LDA and QDA).

Table 7. Mean (plus or minus standard deviation) of scores for optimal parameter settings for each of the specified algorithms predicting s_i^{W}†

Algorithm	*Optimal parameters*	*F1*	*Precision*	*Recall*
AB	$n=100$, $d=1$, $\lambda=0.1$, balanced	*0.80 (± 0.11)*	0.82 (± 0.10)	0.78 (± 0.12)
GB	$n=200$, $d=1$, $\lambda=0.1$	0.79 (± 0.10)	0.80 (± 0.09)	0.78 (± 0.12)
RF	$n=200$, $d=1$, balanced	0.78 (± 0.10)	*0.85 (± 0.14)*	0.76 (± 0.16)
kNN	$k=35$	0.76 (± 0.09)	0.81 (± 0.04)	0.73 (± 0.17)
RBFSVC	$C=1$, $\gamma=0.1$, balanced	0.76 (± 0.11)	0.78 (± 0.08)	0.76 (± 0.18)
LR	$C=1$, L1-penalty	0.75 (± 0.12)	0.76 (± 0.10)	0.76 (± 0.18)
LinSVC	$C=0.01$	0.74 (± 0.10)	0.77 (± 0.07)	0.73 (± 0.17)
LDA		0.74 (± 0.13)	0.69 (± 0.14)	*0.81 (± 0.17)*
MNB	$\alpha=10^{-10}$	0.70 (± 0.12)	0.71 (± 0.15)	0.71 (± 0.12)
QDA		0.67 (± 0.15)	0.63 (± 0.12)	0.73 (± 0.21)

†The scoring function F1 (used to rank the results) is used to optimize across the parameter settings in the parameter grid. Each parameter setting is evaluated by using stratified fivefold cross-validation. In each column the maximum score is highlighted.

more difficult to estimate the industry class by websites than by the business register, leading to the considerable differences between the results of the two estimations. Hence, in future work it might be worthwhile to obtain more training data to improve the results.

The final classification algorithm that we use to estimate s_i^{W} is RF, with parameters $n=200$, $d=1$ and the balanced class weighting scheme (see Table 7). The reason for this choice is that RF maximizes mean precision. Moreover, the local behaviour of the F1-score of RF, as a function of the algorithm parameters, is more stable around the optimal parameters compared with the local behaviour for AB and GB (see Appendix C).

4.3. Estimating cross-border Internet purchases

In Table 8, we present our final estimates of cross-border Internet purchases within the EU by Dutch consumers. Recall that the algorithm that was chosen in Section 4.1 has now been (re)trained on the entire training data set indexed by I_{M}. It resulted in a model $\hat{s}_i^{\mathrm{BR}}$ that was qualified to predict s_i on the remaining part of the data set of tax returns, indexed by $I \backslash I_{\mathrm{M}}$. Similarly, a model $\hat{s}_i^{\mathrm{W}}$ has been trained by using the algorithm that was chosen in Section 4.2. The two models were combined into a final model $\hat{s}_i$ (as described in Section 3.1.3; see also Fig. 2). The comparison between the model $\hat{s}_i$ and the true observed values s_i on the test data set yields the values TP = 8, FP = 4, TN = 62 and FN = 5. It follows that

$$\hat{P}=\begin{pmatrix} 8/13 & 5/13 \\ 4/66 & 62/66 \end{pmatrix} \approx \begin{pmatrix} 0.615 & 0.385 \\ 0.061 & 0.939 \end{pmatrix}.$$

The main results of the paper are shown in Table 8. The values y_{M} contain the total cross-border Internet purchases for companies in the set I_{M}. The categorization for companies in I_{M} has been manually determined and can be considered free from errors. The values $\hat{y}$ contain the additional estimated cross-border Internet purchases for companies in the set $I \backslash I_{\mathrm{M}}$. The values λ_{opt} contain the optimal values of λ in minimizing the mean-squared error of the estimated bias of $\hat{y}$. Note that all optimal values of λ are equal to 0, meaning that the increased variance dominates the decreased squared bias of $\hat{\mathbf{B}}_1$ compared with $\hat{\mathbf{B}}_0$. This is due to the relatively high off-diagonal values in the matrix $\hat{P}$. The values $\hat{B}_{\lambda_{\mathrm{opt}}}$ represent the estimated bias of $\hat{y}$ for the

Table 8. Final results of cross-border Internet purchases within the EU by Dutch consumers

Year	y_{M} *(€ × 10⁶)*	$\hat{y}$ *(€ × 10⁶)*	λ_{opt}	$\hat{B}_{\lambda_{\mathrm{opt}}}$	y *(€ × 10⁶)*	*Std(y)(€ × 10⁶)*
2014	405	495	0	63	837	97
2015	565	586	0	21	1132	101
2016	725	667	0	19	1372	110

optimal value $\lambda = \lambda_{\mathrm{opt}}$. Note that the bias strongly differs across the three years. The values y show the final estimate of the total cross-border Internet purchases, computed as

$$y = y_{\mathrm{M}} + \hat{y} - \hat{B}_{\lambda_{\mathrm{opt}}}.$$

The last column in Table 8 contains the standard deviation of y, estimated as outlined at the end of Section 3.2.

In the Netherlands, total household consumption on retail goods (food and durable goods, codes 1000 up until and including 3000) in 2016 was equal to €87206 million, according to Statistics Netherlands (`https://opendata.cbs.nl`). Statistics Netherlands does not publish the total on-line consumption of goods by Dutch consumers. The only currently available estimate is by Thuiswinkel.org and GfK and it is based on consumer surveys. The estimate of 2016 equals€11.01 billion. It seems possible that just over 12% of on-line consumption by Dutch consumers is spent at foreign webshops that are established within the EU. Besides, Statistics Netherlands does publish year-by-year growth figures on on-line retail sales by Dutch webshops. In 2016, this year-by-year growth was equal to 22.1%. It is quite similar to the growth of 21.2% that we find by comparing the values of y in 2015 and 2016 as presented in Table 8.

Reflecting on our findings, we note that the standard deviation of the final estimate would still be too large for official statistical purposes. However, as will be discussed more thoroughly in Section 4.4, our findings prove to be a significant improvement over currently available alternative estimates.

4.4. Comparison with demand-side approach

In Section 1, we claimed that our data-driven supply-side approach would be more accurate than demand-side approaches to estimate cross-border Internet purchases within the EU. To justify this claim, we compare our results for the Netherlands with the results of the consumer survey approach by market research institute GfK (commissioned by Thuiswinkel.org on behalf of Ecommerce Europe). We choose to use the estimate by these commercial organizations, as, to the best of our knowledge, there is no scientific literature reporting the total cross-border Internet purchases by Dutch consumers.

In 2016, total cross-border on-line consumption by Dutch consumers according to GfK was equal to €637 million, €190 million of which were spent in China and €70 million in the USA. This implies that at most €377 million were spent within the EU, but this figure includes on-line consumption of both goods and services.

Moveover, the fraction of on-line consumption of goods in the total on-line consumption in 2016, as reported by GfK, was €11.01 billion/€20.16 billion = 0.55. We assume that this proportion is independent of the country in which the goods or services were purchased. As a result, cross-border on-line purchases of goods within the EU, according to GfK, would approximately equal €206 million in 2016.

We, however, find €1372 million for 2016 with a standard deviation of €110 million, i.e. 8%. The estimate is more than six times as high as that of GfK. The results show the severe downward bias in using demand-side approaches to estimate cross-border on-line consumption and it motivates the implementation of our approach for other EU member states.

5. Main conclusion and future work

We have proposed a methodology to measure cross-border Internet purchases within the EU by using tax data, a business register and website data. We have implemented data-driven methods to combine these supply-side data sources in a computationally efficient manner. Applied to the Netherlands, the approach leads to a strong improvement of existing approaches that are based on consumer surveys. In particular, market research institute GfK (commissioned by Thuiswinkel.org on behalf of Ecommerce Europe) use consumer surveys and estimated cross-border Internet purchases by Dutch consumers within the EU in 2016 to be approximately €206 million. Our approach yields an estimate of €1372 million, i.e. six times as high as GfK's estimate, with a standard deviation of €110 million (8%).

The approach that we propose requires foreign companies' tax returns to contain only the legal company name and the turnover from sales of goods to consumers. Because of EU VAT legislation these data are available in every EU member state. In fact, we do not require the economic activity of a company to be accurately available in filed tax returns. The training and test set could even be constructed without any known economic activity, by viewing all companies as belonging to the same class and following the construction that was described in Sections 2.2 and 2.3. We also do not assume that the URL of the home page of a company is available in filed tax returns. Moreover, the additional data (the business register and websites) that are required by the approach proposed are open data sources. Hence, our main conclusion is that the approach is applicable in any EU member state and more accurately estimates cross-border Internet purchases within the EU.

In addition to our methodological contribution to official statistics concerning cross-border Internet purchases, we point out two aspects of our contribution that might be of interest to a general audience in statistics. The first aspect is our fully data-driven (and therefore generic) approach for probabilistic record linkage of firm level data sources. The novelty of our approach compared with the extant literature is that we use machine learning to maximize the accuracy of the record linkage. In this regard, we improve on existing methods as the optimizations (choosing an optimal string matching algorithm and similarity threshold) are fully automated. Moreover, our approach is computationally efficient by using state of the art hashing techniques from computer science. Therefore, our approach can be applied to related large-scale (text-based) probabilistic record linkage problems. The second aspect is the observation that aggregation (e.g. summing) after running a classification algorithm yields (potentially strongly) biased estimates. We believe that we are the first to make this observation in the field of machine learning. As a first step, we have shown

(a) that, in many fields outside machine learning, techniques have been developed to correct the bias of aggregate estimates and
(b) that we can directly apply the techniques to classification algorithms.

In our view, the bias of aggregates based on results from classification algorithms deserves more investigation beyond our first step.

Although our new methodology improves the estimation of cross-border Internet purchases within the EU, we point out two potential sources of bias of our current approach. First, companies with sales below the threshold value in the country of destination (in the Netherlands: €100000) do not have to file a tax return. The Internet purchases for such small companies are therefore missing in an estimation based on tax data. Yet, the sales of small companies via marketplaces (e.g. Amazon) are included in tax data. Second, the reported turnover from sales to consumers might be inaccurate, potentially leading to an underestimation of total cross-border Internet purchases. However, this underestimation is expected to be minimal because of strict law enforcement by, and collaboration between, tax authorities in the EU. We have not aimed to correct for either of these two biases, as no data are available to estimate them. Moreover, we aimed to show the downward bias of consumer survey approaches compared with a supply-side approach in estimating cross-border Internet purchases within the EU. We therefore do not mind if our supply-side approach still yields a conservative estimate.

Future work on measuring cross-border Internet purchases within the EU might focus on improving the predictions by websites of company classifications. The empirical results show that this is the weakest part of the approach that we propose, as the F1-scores for website-based predictions are lower than the F1-scores for the predictions based on using the business register. The results may be improved by enlarging the training set of the URL retrieval software from Dutch to European websites by using the company names and URLs that are registered in the business register. We consider this improvement outside the scope of the current paper, as the results of our data-driven supply-side approach already show a strong improvement compared with existing consumer survey approaches.

Finally, further applications of the supply-side approach proposed include revealing the structure of the cross-border on-line retail market in any EU member state. Our approach directly returns a list of foreign webshops and their annual cross-border Internet sales to the observing member state. If the information on domestic webshops that are active within the member state's e-commerce market is complemented, the structure of that market may be analysed. Related to this is the export of the webshops that are established in a single EU member state, being the supply-side counterpart of cross-border on-line consumption within a member state. It might be interesting to compare the two market structures within individual member states and to compare the market structures between member states within the EU.

Acknowledgements

We express our sincere gratitude to Arjan van Loon for many fruitful discussions on data analysis and research methods. Moreover, we are grateful to Dick Windmeijer and Guido van den Heuvel for developing the URL retrieval and web scraping software and running it on the data. Furthermore, we thank Myra Bissumbhar and Marja Groen for manually categorizing companies in the training and test data sets. We are grateful to Willem de Jong and Jos Erkens for helpful discussions on EU VAT legislation and we thank Javier Garcia-Bernardo for posing the idea of hashing and combining string matching methods for probabilistic record linkage. Finally, we profoundly thank the reviewers for their comments on an earlier version of the paper. The research was funded by Statistics Netherlands and the Dutch Ministry of Economic Affairs and Climate Policy.

The views that are expressed in this paper are those of the authors and do not necessarily reflect the policy of Statistics Netherlands.

Appendix A: Estimating the industry class by the business register

This appendix contains the details of the first three steps of our four-step approach for estimating the industry class by the business register. The details of step IV can be found in the main text.

A.1. Step I: stemming company names

The stemming of company names in the tax data and the business register follows the following three steps, inspired by Lovins (1968) and Porter (1980).

Step 1: use the business register to create, for each country in the EU and each $n = 1, 2, 3, 4$, a list of the five most common legal company name *suffixes* (i.e. end-of-string words) of length n. Complement the list with the types of business entities from Table 9.
Step 2: for each EU member state, identify the *prefixes* of the suffixes in the list obtained in step 1 as well as the *suffix class* (business type–member state). Concatenate the suffix–prefix lists obtained into a single list.
Step 3: for each legal company name, search each of its words (starting from the second word) in the suffix–prefix list from step 2. Stop if a match is found. Split the name into a *stem* and a suffix, storing its suffix class.

We include an example to illustrate the stemming procedure. In Germany, the most common type of business entity is *Gesellschaft mit beschränkter Haftung* (GmbH), which is similar to a private company limited by shares (LTD). This type of business entity will show up in step 1 for `country = 'DE'` and `n = 4`. Many variations may occur due to partial abbreviations (e.g. *Gesellschaft mbH*). Step 2 ensures that only three suffix–prefixes must be searched for in step 3: GMBH, G M B H and Gesellschaft. The *suffix class* corresponding to each of these three suffix–prefixes is ‘private company limited by shares, German’ (LTD–DE). Now, take as an example a German company named *Muller GmbH*, which is stored as *muller gmbh* after the preprocessing step. The algorithm starts searching the second word, `gmbh`, in the suffix–prefix list. It is found, and three values are stored for this company: `stem = 'muller'`, `suffix = 'gmbh'`, and `suffix_class = 'LTD/DE'`. In general, if the second word is not found, the algorithm would continue

Table 9. Overview of the types of business entities per EU member state, obtained from *Wikipedia* (https://en.wikipedia.org/wiki/List_of_business_entities)

Country	*Code*	*Types of business entities*
Austria	AT	AG, GmbH, KG, GmbH & Co. KG
Belgium	BE	BVBA, NV, SA
Bulgaria	BG	AD, EAD, EOOD, OOD
Croatia	HR	d.d., d.o.o.
Cyprus	CY	(Same as GB)
Czech Republic	CZ	a.s., s.r.o.
Denmark	DK	ApS, A/S, A.M.B.A.
Estonia	EE	OÜ, AS
Finland	FI	Oy, oyj
France	FR	SARL, SA
Germany	DE	OHG, KG, AG, GmbH, GmbH & Co. KG/AG/OHG
Greece	GR	A.E., E.P.E.
Hungary	HU	*korlatolt felelossegu tarsasag*, (*nyilvanosan/zartkozuen mukodo*) *reszvenytarsasag*
Ireland	IE	(Same as GB)
Italy	IT	s.r.l., s.p.a., *societa a responsabilita limitata*
Latvia	LV	SIA, AS
Lithuania	LT	UAB, AB
Luxembourg	LU	S.A., S.A.R.L., SECS
Malta	MT	(Same as GB)
Netherlands	NL	BV, NV
Poland	PL	Sp. Z.O.O., S.A.
Portugal	PT	lda., S.A.
Romania	RO	S.R.L., S.A.
Slovakia	SK	S.R.O., A.S.
Slovenia	SI	d.d., d.o.o.
Spain	ES	S.A, *sociedad anonima*, S.L., *sociedad limitada*
Sweden	SE	AB, *aktiebolag*
UK	GB	private limited company, ltd, limited, public limited company, plc

with the third word, until the name's final word. If no matches are found at all, the stem equals the name: `suffix =''` and `suffix_class = ''` (empty strings).

A.2. Step II: locality-sensitive hashing

The tax data and the business register do not contain characters with diacritical marks (e.g. the German umlaut as in 'ü'). A German name such as Müller (English: Miller) has been registered by using plain alphabetic characters instead. For the 'ü' in Müller, two conventions exist: Muller or Mueller. This leads to potential spelling differences between the tax data and the business register for the same company. Another common difference is the use or omission of spaces (e.g. webshop *versus* web shop).

As discussed in the main text, we use the famous LSH scheme MinHash (Broder, 1997) to match tax data and the business register in an elegant and efficient way concerning approximate string matching. The following four paragraphs elaborate on

(a) the approximate string matching method for which MinHash is locality sensitive,
(b) creating the MinHash signatures,
(c) creating the LSH Forest data structure and
(d) the details of our implementation in Python.

The LSH scheme MinHash is locality sensitive for the Jaccard distance on character n-grams, or n-shingles (Leskovec *et al.* (2014), chapter 3). A *character n-gram* is defined as a substring of n consecutive characters in a string. As an example, the set of character 3-grams, or trigrams, of the string 'webshop' is the set {'web', 'eb ', 'b s', ' sh', 'sho', 'hop'}. For $n \in \mathbb{N}$, write f_n for the functions mapping a string to its set of character n-grams. The Jaccard distance between two sets A and B is defined as

$$d_{\mathrm{J}}(A, B) = 1 - |A \cap B|/|A \cup B|.$$

The n-gram Jaccard distance $d_{\mathrm{J},n}$ between two strings s and t is defined as

$$d_{\mathrm{J},n}(s, t) = 1 - d_{\mathrm{J}}\{f_n(s), f_n(t)\}.$$

For MinHash, a string is identified by a binary-valued vector in $\{0, 1\}^{c_n}$, with $c_n = (26 + 10 + 1)^n$ (enumerated in the order of the alphabet (26), digits (10) and space (1)). In fact, a string is stored only as the list of index numbers (according to the n-gram enumeration) of the n-grams that it contains. The randomized dimensionality reduction MinHash computes a k-bit min-hash signature for each f_n^a in the following way. First, randomly choose k hash functions $h_1, \ldots, h_k$ from the family of random linear functions of the form $h(x) = (\alpha x + \beta) \bmod p$, with a and b integers and p a fixed, large prime number. Then, randomly choose k hash functions $g_1, \ldots, g_k$ mapping the values $0, \ldots, p-1$ uniformly at random onto $\{0, 1\}$. The jth bit of the k-bit min-hash signature of a is then given by $g_j[\min_i\{h_j(v_i)\}]$, where v is the list containing the index numbers of the character n-grams of a.

To reduce the number of evaluations of $d_{\mathrm{J},n}$ that are required to match the two data sources, we apply the LSH Forest data structure (Bawa *et al.*, 2005) on the results of MinHash. In short, an *LSH tree* is defined as the logical prefix tree on all k-bit signatures. The LSH forest consists of l LSH trees, each constructed with an independently drawn random sequence of hash functions from the described family of hash functions (MinHash). Now, given the stem s of a company name from the tax data, each of the l LSH trees is updated with an additional leaf node containing (the end point of the path through the LSH tree specified by the k-bit signature of) $s(a)$. The LSH trees are then searched bottom up simultaneously, starting from the new leaf node, until the m most similar items are identified. Consult Bawa *et al.* (2005) for further algorithmic details.

In our implementation in Python, the function `MinHashLSHForest` from the Python library `datasketch` is used (`https://github.com/ekzhu/datasketch`). We set $n = 3$, i.e. we consider the Jaccard distance of character trigrams. The total number of hash functions is fixed to be 64 and the number of LSH trees was set to the default value $l = 8$. The datasketch implementation then fixes $k = 64/8 = 8$ for the length of the min-hash signatures that are used to build each of the LSH trees. The choice of $k = 8$ is relatively small but works already quite well in our case, as shown in Section 4.1. The top $m = 100$ most similar leaf nodes (stems of company names from the business register) from the LSH forest are returned for each stem of company names from the tax data. If the suffix class of a company from the business register is different from that of the company from the tax data, the company from the business register is removed from the list. The resulting lists serve as input for the next part of our data-driven approach for firm level record linkage.

A.3. Step III: combining string distance metrics

We combine the following eight commonly used string distance metrics:

(a) 1, the normalized Levenshtein (or edit) distance;
(b) 2, the Jaro–Winkler divergence (not a metric in the mathematical sense (Winkler, 1990));
(c) 3–5, the Jaccard distance on sets of character 1-, 2- and 3-grams;
(d) 6–8, the cosine distance on term frequency vectors of character 1-, 2- and 3-grams.

The Jaro–Winkler, Jaccard and cosine distances are defined as 1 minus the corresponding string similarity measures and always take values in the interval [0,1]. The Levenshtein distance is normalized to the interval [0,1], which is achieved by dividing by the maximum length of the two input strings. All metrics were defined and compared by Cohen *et al.* (2003). For a more recent discussion, see Leskovec *et al.* (2014), pages 87–93.

At the end of this step, each company in the tax data is equipped with an eight-dimensional vector containing values in the interval [0,1]. These values can be interpreted as the distance (along different metrics) to the set of EU retail companies in the business register. The values will be used as features in the machine learning algorithms as described in Section 3.1.1.4.

Appendix B: Finding λ^*

This appendix describes how the optimal value $\lambda=\lambda^*$ is found. Recall that Van Delden *et al.* (2016) considered the linear combinations $\hat{\mathbf{B}}_\lambda=(1-\lambda)\hat{\mathbf{B}}_0+\lambda\hat{\mathbf{B}}_1$ for $\lambda\in[0,1]$ of the bias estimators given by equations (8) and (9). They proposed to find the optimal value $\lambda=\lambda^*$ that minimizes the mean-squared error of the first component $(\hat{\mathbf{B}}_\lambda)_1$ of $\hat{\mathbf{B}}_\lambda$ as an estimator of the bias $\mathbf{B}(\hat{\mathbf{y}})$. This mean-squared error is given by

$$\text{mse}\{(\hat{\mathbf{B}}_\lambda)_1\}=\{\mathbf{B}(\hat{\mathbf{B}}_\lambda)\}_1^2+\{V(\hat{\mathbf{B}}_\lambda)\}_{11},$$

where $\{V(\hat{\mathbf{B}}_\lambda)\}_{11}$ denotes the upper left-hand entry in the variance–covariance matrix of the 2-vector $\hat{\mathbf{B}}_\lambda$. The following iterative approach was suggested by Van Delden *et al.* (2016) to find λ^*.

Step 1: initialize—start with $\lambda_{\text{old}}=0$.

Step 2: compute $\hat{\mathbf{B}}=\hat{\mathbf{B}}_0$ and $\hat{\Omega}=\hat{V}(\hat{\mathbf{y}})$.

Step 3: compute $\lambda_{\text{new}}=\max\{0, \min\{1, (m_1-m_3+m_4)/(m_1+m_2-2m_3+m_4)\}\}$, where

$$\begin{aligned}m_1&=((\hat{P}^{\mathrm{T}}-I_2)\hat{\mathbf{B}})_1^2,\\ m_2&=((\hat{P}^{\mathrm{T}}-I_2)\hat{\Omega}(\hat{P}^{\mathrm{T}}-I_2)^{\mathrm{T}}\hat{Q}^{\mathrm{T}}\hat{Q})_{11},\\ m_3&=(\tfrac{1}{2}(\hat{P}^{\mathrm{T}}-I_2)\hat{\Omega}(\hat{P}^{\mathrm{T}}-I_2)^{\mathrm{T}}(\hat{Q}+\hat{Q}^{\mathrm{T}}))_{11},\\ m_4&=((\hat{P}^{\mathrm{T}}-I_2)\hat{\Omega}(\hat{P}^{\mathrm{T}}-I_2)^{\mathrm{T}})_{11}.\end{aligned}$$

Step 4: if $|\lambda_{\text{new}}-\lambda_{\text{old}}|<10^{-6}$, stop and return λ_{new}. Otherwise, set

$$\hat{\mathbf{B}}=\hat{\mathbf{B}}_{\lambda_{\text{new}}}=(1-\lambda_{\text{new}})\hat{\mathbf{B}}_0+\lambda_{\text{new}}\hat{\mathbf{B}}_1=(I_2+\lambda_{\text{new}}(\hat{Q}-I_2))\hat{\mathbf{B}}_0.$$

Step 5: set $\lambda_{\text{old}}:=\lambda_{\text{new}}$ and return to step 3.

Details of the derivation of the formulae in step 3 can be found in appendix A3 in Van Delden *et al.* (2015). We indicate that the above iterative procedure is performed for each of the years 2014, 2015 and 2016 separately. The same (estimated) matrix $\hat{P}$ is used for each year. The optimal value of λ might differ across years, as it depends on the annual turnover.

Appendix C: Local behaviour around optimal parameters

This appendix contains additional results on the local behaviour of the mean and standard deviation of F1-scores (obtained from the fivefold cross-validation) around the optimal parameters for the optimal algorithms. The results for $\hat{s}_i^{\mathrm{BR}}$ (by the business register) and $\hat{s}_i^{\mathrm{W}}$ (by websites) are presented separately.

C.1. Business register

Table 10 shows that the results for $C \geqslant 10$ hardly depend on the class weighting scheme chosen. Moreover, different choices of γ and different choices of C, given $C \geqslant 10$, only minimally affect the mean F1-score over the five folds. The standard deviation is similar in each of these parameter settings as well. Thus, the mean F1-score is stable around the optimal parameter setting.

C.2. Websites

Next, we examine the local behaviour of the mean and standard deviation of F1-scores obtained from the fivefold cross-validation around the optimal parameters for the algorithms AB, GB and RF trained to predict $\hat{s}_i^{\mathrm{W}}$.

Starting with the AB algorithm, Table 11 shows that increasing the maximum tree depth d from the optimal value $d=1$ negatively impacts the goodness of fit as measured by F1. Moreover, for $\lambda=0.01$ and $\lambda=1$, the mean F1-score is substantially lower for $d=1$ compared with the optimal $\lambda=0.1$ (all using the balanced class weighting scheme). Thus, the results by the AB algorithm are not stable around the optimal maximum tree depth $d=1$ and not around the optimal learning rate $\lambda=0.1$. The results are less sensitive to the choice of the class weighting scheme, for $\lambda=0.1$.

Studying Table 12, we may conclude that the mean F1-scores of the GB algorithm are not very stable around the optimal parameter setting ($n=200, d=1, \lambda=0.1$). Increasing the maximum depth d from the

Table 10. Mean (and standard deviation of) F1-scores for the RBFSVC algorithm trained to predict $\hat{s}_i^{\mathrm{BR}}$†

γ	*Results for uniform class weighting*			*Results for balanced class weighting*		
	C = 1	*C = 10*	*C = 100*	*C = 1*	*C = 10*	*C = 100*
0.001	—	0.93 (± 0.05)	0.94 (± 0.02)	—	0.91 (± 0.04)	0.94 (± 0.02)
0.01	0.93 (± 0.05)	0.94 (± 0.02)	0.92 (± 0.03)	0.92 (± 0.05)	0.94 (± 0.02)	0.92 (± 0.03)
0.1	0.94 (± 0.02)	0.93 (± 0.02)	0.96 (± 0.04)	0.94 (± 0.02)	0.93 (± 0.02)	0.96 (± 0.04)
1	0.93 (± 0.03)	0.95 (± 0.04)	*0.97 (± 0.03)*	0.94 (± 0.02)	0.95 (± 0.04)	0.97 (± 0.03)

†The optimal parameter setting ($C=100, \gamma=1$, uniform) with corresponding F1-score 0.97 is displayed in italics.

Table 11. Mean (and standard deviation of) F1-scores for the AB algorithm trained to predict $\hat{s}_i^{\mathrm{W}}$†

n	*Results for $\lambda=0.01$, balanced class weighting*			*Results for $\lambda=0.1$, uniform class weighting*		
	d=1	*d=2*	*d=3*	*d=1*	*d=2*	*d=3*
50	0.77 (± 0.06)	0.75 (± 0.13)	0.67 (± 0.15)	0.74 (± 0.14)	0.70 (± 0.17)	0.69 (± 0.12)
100	0.75 (± 0.12)	0.73 (± 0.14)	0.69 (± 0.14)	0.78 (± 0.13)	0.75 (± 0.12)	0.70 (± 0.15)
200	0.75 (± 0.12)	0.71 (± 0.15)	0.67 (± 0.16)	0.77 (± 0.09)	0.73 (± 0.14)	0.67 (± 0.16)
500	0.78 (± 0.10)	0.69 (± 0.15)	0.70 (± 0.15)	0.75 (± 0.10)	0.76 (± 0.13)	0.69 (± 0.15)
	$\lambda=0.1$, balanced class weighting			*$\lambda=1$, balanced class weighting*		
50	0.80 (± 0.09)	0.74 (± 0.15)	0.67 (± 0.12)	0.72 (± 0.08)	0.71 (± 0.06)	0.71 (± 0.14)
100	*0.80 (± 0.11)*	0.75 (± 0.14)	0.65 (± 0.09)	0.71 (± 0.08)	0.71 (± 0.06)	0.70 (± 0.11)
200	0.79 (± 0.10)	0.72 (± 0.13)	0.67 (± 0.09)	0.70 (± 0.07)	0.71 (± 0.10)	0.70 (± 0.11)
500	0.75 (± 0.09)	0.73 (± 0.12)	0.68 (± 0.14)	0.72 (± 0.06)	0.69 (± 0.14)	0.66 (± 0.10)

†The optimal parameter setting ($n=100, d=1, \lambda=0.1$, balanced) with corresponding F1-score 0.80 is displayed in italics.

Table 12. Mean (and standard deviation of) F1-scores for the GB algorithm trained to predict $\hat{s}_i^{W}$†

n	*Results for λ = 0.1 (d varies)*			*Results for d = 1 (λ varies)*		
	d = 1	*d = 2*	*d = 3*	*λ = 0.01*	*λ = 0.1*	*λ = 1*
50	0.75 (± 0.12)	0.72 (± 0.14)	0.71 (± 0.15)	0.65 (± 0.05)	0.75 (± 0.12)	0.74 (± 0.10)
100	0.77 (± 0.12)	0.71 (± 0.15)	0.69 (± 0.13)	0.71 (± 0.08)	0.77 (± 0.12)	0.75 (± 0.10)
200	*0.79 (± 0.10)*	0.71 (± 0.15)	0.69 (± 0.13)	0.73 (± 0.12)	*0.79 (± 0.10)*	0.73 (± 0.10)
500	0.74 (± 0.12)	0.72 (± 0.16)	0.66 (± 0.14)	0.76 (± 0.12)	0.74 (± 0.12)	0.73 (± 0.10)

†The optimal parameter setting ($n = 200, d = 1, \lambda = 0.1$) with corresponding F1-score 0.79 is displayed in italics. Note that the parameter settings for the second and sixth column are identical.

Table 13. Mean (and standard deviation of) F1-scores for the RF algorithm trained to predict $\hat{s}_i^{W}$†

n	*Results for uniform class weighting*			*Results for balanced class weighting*		
	d = 1	*d = 2*	*d = 3*	*d = 1*	*d = 2*	*d = 3*
50	0.75 (± 0.13)	0.72 (± 0.14)	0.74 (± 0.15)	0.74 (± 0.13)	0.76 (± 0.10)	0.71 (± 0.16)
100	0.76 (± 0.13)	0.73 (± 0.14)	0.69 (± 0.17)	0.76 (± 0.13)	0.75 (± 0.13)	0.68 (± 0.15)
200	0.73 (± 0.14)	0.73 (± 0.14)	0.67 (± 0.16)	*0.78 (± 0.10)*	0.71 (± 0.13)	0.71 (± 0.13)
500	0.76 (± 0.14)	0.67 (± 0.14)	0.68 (± 0.13)	0.75 (± 0.10)	0.70 (± 0.15)	0.71 (± 0.12)

†The optimal parameter setting ($n = 200, d = 1$, balanced) with corresponding F1-score 0.78 is displayed in italics.

optimal value $d = 1$ while fixing the optimal learning rate $\lambda = 0.1$ leads to a drop in mean F1-score. The same holds for changing the optimal learning rate $\lambda = 0.1$ while fixing $d = 1$.

Finally, we present the results of the RF algorithm on the training data set in Table 13. For the balanced class weighting scheme, the results seem stable as n increases. Moreover, the variance is smaller than in the uniform class weighting scheme. However, increasing the maximum depth d from the optimal value $d = 1$ leads to a drop in mean F1-score.

References

Autor, D., Dorn, D., Hanson, G. H., Pisano, G. and Shu, P. (2017) Foreign competition and domestic innovation: evidence from U.S. patents. *Working Paper 22879*. National Bureau of Economic Research, Cambridge.

Bailey, J., Gao, G., Jank, W., Lin, M., Lucas, H. C. and Viswanathan, S. (2008) The long tail is longer than you think: the surprisingly large extent of online sales by small volume sellers. *Scholarly Paper 1132723*. Social Science Research Network, Rochester.

Balsmeier, B., Assaf, M., Chesebro, T., Fierro, G., Johnson, K., Johnson, S., Li, G.-C., Lück, S., O'Reagan, D., Yeh, B., Zang, G. and Fleming, L. (2018) Machine learning and natural language processing on the patent corpus: data, tools, and new measures. *J. Econ. Mangmnt Strat.*, **27**, 535–553.

Bawa, M., Condie, T. and Ganesan, P. (2005) LSH Forest: self-tuning indexes for similarity search. In *Proc. 14th Int. Conf. World Wide Web, Chiba* (eds A. Ellis and T. Hagino), pp. 651–660. New York: Association for Computing Machinery.

Bena, J., Ferreira, M. A., Matos, P. and Pires, P. (2017) Are foreign investors locusts?: The long-term effects of foreign institutional ownership. *J. Finan. Econ.*, **126**, 122–146.

Blazquez, D., Domenech, J., Gil, J. A. and Pont, A. (2018) Monitoring e-commerce adoption from online data. *Knowledge and Information Systems*.

Breiman, L. and Spector, P. (1992) Submodel selection and evaluation in regression: the X-random case. *Int. Statist. Rev.*, **60**, 291–319.

Broder, A. Z. (1997) On the resemblance and containment of documents. In *Proc. Compression and Complexity of Sequences, Salerno* (eds B. De Santis, U. Vaccaro and J. A. Storer), pp. 21–29. Washington DC: Institute of Electrical and Electronics Engineers Computer Society.

Cardona, M. and Duch-Brown, N. (2016) Delivery costs and cross-border e-commerce in the EU Digital Single Market. *Working Paper on Digital Economy 2016/03*. Joint Research Centre, Seville.

Cohen, W., Ravikumar, P. and Fienberg, S. (2003) A comparison of string metrics for matching names and records. In *Proc. Int. Conf. Information Integration on the Web, Acapulco* (eds S. Kambhampati and C. A. Knoblock), pp. 73–78. Palo Alto: American Association for Artificial Intelligence.

Davis, J. and Goadrich, M. (2006) The relationship between precision-recall and ROC curves. In *Proc. 23rd Int. Conf. Machine Learning* (eds W. Cohen and A. Moore), pp. 233–240. New York: Association for Computing Machinery.

European Commission (2010) Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee of the Regions—a digital agenda for Europe, COM/2010/0245. Publication Office of the European Union, Luxembourg.

European Commission (2015) Monitoring the digital economy & society 2016-2021. European Commission, Luxembourg. (Available from `http://ec.europa.eu/newsroom/dae/document.cfm?doc_id=13706`.)

Fellegi, I. P. and Sunter, A. B. (1969) A theory for record linkage. *J. Am. Statist. Ass.*, **64**, 1183–1210.

Garcia-Bernardo, J. and Takes, F. W. (2018) The effects of data quality on the analysis of corporate board interlock networks. In *Informn Syst.*, **78**, 164–172.

Gomez-Herrera, E., Martens, B. and Turlea, G. (2014) The drivers and impediments for cross-border e-commerce in the EU. *Inform. Econ. Poly*, **28**, 83–96.

Hall, B. H., Jaffe, A. B. and Trajtenberg, M. (2001) The NBER patent citation data file: lessons, insights and methodological tools. *Working Paper 8498*. National Bureau of Economic Research, Cambridge.

Han, J., Kamber, M. and Pei, J. (2011) *Data Mining: Concepts and Techniques*, 3rd edn. Waltham: Morgan Kaufmann.

Hastie, T., Tibshirani, R. and Friedman, J. (2009) *The Elements of Statistical Learning*, 2nd edn. New York: Springer.

Jeni, L. A., Cohn, J. F. and De La Torre, F. (2013) Facing imbalanced data—recommendations for the use of performance metrics. In *Proc. Conf. Affective Computing and Intelligent Interaction, Geneva* (eds T. Pan, C. Pelachaud and N. Sebe), pp. 245–251. Washington DC: Institute of Electrical and Electronics Engineers Computer Society.

Kohavi, R. (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proc. Int. Jt Conf. Artificial Intelligence, Montreal* (ed. C. R. Perrault), pp. 1137–1145. San Francisco Morgan Kaufmann.

Lash, T. L., Fox, M. P. and Fink, A. K. (2009) *Applying Quantitative Bias Analysis to Epidemiologic Data*. New York: Springer.

Leskovec, J., Rajaraman, A. and Ullman, J. D. (2014) *Mining of Massive Datasets*. Cambridge: Cambridge University Press.

Lovins, J. B. (1968) Development of a stemming algorithm. *Mech. Transl. Computnl Ling.*, **11**, 22–31.

Löw, F., Knöfel, P. and Conrad, C. (2015) Analysis of uncertainty in multi-temporal object-based classification. *J. Photgramm. Remote Sens.*, **105**, 91–106.

Ma, S., Chai, Y. and Zhang, H. (2018) Rise of cross-border e-commerce exports in China. *China Wrld Econ.*, **26**, 63–87.

Manning, C. D., Raghavan, P. and Schütze, H. (2009) *Introduction to Information Retrieval*. Cambridge: Cambridge University Press.

Marcus, J. S. and Petropoulos, G. (2016) E-commerce in Europe: parcel delivery prices in a digital single market. *Policy Contribution 2016/09*. Breugel, Brussels.

Martikainen, E., Schmiedel, H. and Takalo, T. (2015) Convergence of European retail payments. *J. Bankng Finan.*, **50**, 81–91.

Minges, M. (2016) In search of cross-border e-commerce trade data. United Nations Conference on Trade and Development, Geneva. (Available from `http://unctad.org/en/PublicationsLibrary/tn_unctad_ict4d06_en.pdf`.)

Oestreicher-Singer, G. and Sundararajan, A. (2012) Recommendation networks and the long tail of electronic commerce. *Mangmnt Informn Syst. Q.*, **36**, 65–83.

Porter, M. F. (1980) An algorithm for suffix stripping. *Program*, **14**, no. 3, 130–137.

Ribeiro, S. P., Menghinello, S. and De Backer, K. (2010) The OECD ORBIS database: responding to the need for firm-level micro-data in the OECD. *Statistics Working Paper 2010/1*. Organisation for Economic Co-operation and Development, Paris.

Schu, M. and Morschett, D. (2017) Foreign market selection of online retailers—a path-dependent perspective on influence factors. *Int. Bus. Rev.*, **26**, 710–723.

Tarasconi, G. and Menon, C. (2017) Matching Crunchbase with patent data. *Science, Technology and Industry Working Paper 2017/07*. Organisation for Economic Co-operation and Development, Paris.

Ten Bosch, O. and Windmeijer, D. (2018) Web scraping enterprise statistics. *ESSNET Big Data Work Pack-*

age 2 Deliverable 2.4 Final Report, pp. 41–44. Eurostat, Luxembourg. (Available from `https://webgate.ec.europa.eu/fpfis/mwikis/essnetbigdata/images/e/ee/Wp2_Del2_4.pdf`.)

Van Delden, A., Scholtus, S. and Burger, J. (2015) Quantifying the effect of classification errors on the accuracy of mixed-source statistics. *Discussion Paper 2015–10*. Statistics Netherlands, The Hague. (Available from `https://www.researchgate.net/publication/281450992_Quantifying_the_effect_of_classification_errors_on_the_accuracy_of_mixed-source_statistics`.)

Van Delden, A., Scholtus, S. and Burger, J. (2016) Accuracy of mixed-source statistics as affected by classification errors. *J. Off. Statist.*, **32**, 619–642.

Winkler, W. E. (1990) String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage. *Proc. Surv. Res. Meth. Sect. Am. Statist. Ass.*, 354–359.

Witten, I. H., Frank, E., Hall, M. A. and Pal, C. J. (2017) *Data Mining: Practical Machine Learning Tools and Techniques*, 4th edn. Cambridge: Morgan Kaufmann.