HHS Public Access

Author manuscript

Stat J IAOS. Author manuscript; available in PMC 2021 August 18.

Published in final edited form as:

Stat J IAOS. 2021 June 3; 37(2): 673–680. doi:10.3233/sji-200779.

Using supervised machine learning to identify efficient blocking schemes for record linkage

Scott R. Campbell[a,*], **Dean M. Resnick**[a], **Christine S. Cox**[a], **Lisa B. Mirel**[b]

[a]NORC at the University of Chicago, Bethesda, MD, USA

[b]Centers for Disease Control and Prevention, National Center for Health Statistics, Hyattsville, MD, USA

Abstract

Record linkage enables survey data to be integrated with other data sources, expanding the analytic potential of both sources. However, depending on the number of records being linked, the processing time can be prohibitive. This paper describes a case study using a supervised machine learning algorithm, known as the Sequential Coverage Algorithm (SCA). The SCA was used to develop the join strategy for two data sources, the National Center for Health Statistics' (NCHS) 2016 National Hospital Care Survey (NHCS) and the Center for Medicare & Medicaid Services (CMS) Enrollment Database (EDB), during record linkage. Due to the size of the CMS data, common record joining methods (i.e. blocking) were used to reduce the number of pairs that need to be evaluated to identify the vast majority of matches. NCHS conducted a case study examining how the SCA improved the efficiency of blocking. This paper describes how the SCA was used to design the blocking used in this linkage.

Keywords

National Center for Health Statistics; Centers for Medicare & Medicaid Services; National Hospital Care Survey; record linkage; blocking; machine learning

1. Introduction

Within the past decade, interest in the benefits and utility of machine learning (ML) has been rapidly growing [1]. ML algorithms that efficiently process large data have replaced time-consuming ad hoc data processing techniques. ML techniques such as the method described by Giang which uses the Probably Approximately Correct (PAC) learning theory [2] and

[*]Corresponding author: Scott R. Campbell, NORC at the University of Chicago, 4350 East-West Highway, 8th Floor, Bethesda, MD, 20814, USA. Tel.: +1 301 634 9431; Campbell-Scott@norc.org.

Conflict of interest

The authors have no conflicts of interest to report.

Availability of data and material

Researchers who wish to obtain access to the linked 2016 NHCS to 2016/2017 CMS file must submit and have an approved research proposal to the NCHS Research Data Center (RDC): https://www.cdc.gov/rdc/index.htm.

Code availability

Code for this analysis is available upon request

the Sequential Coverage Algorithm (SCA) described by Michelson and Knoblock [3] have been developed to improve the efficiency of record linkage processes. In general, to perform record linkage, a set of potential links are produced by joining two data sources. These potential links are then further processed by a set of deterministic rules and/or probabilistic methods. Common strategies used to join data sources are typically developed ad-hoc (i.e. for a particular task at hand rather than for general usage), which require the judgment of a professional and result in extended processing time. Supervised ML[1] algorithms, however, can be designed to quickly develop a strategy that that accurately and efficiently join the data sources in a timely manner still capturing the vast majority of true matches.

Record linkage enables survey data to be linked to other data sources, expanding the analytic potential of both sources of data. Through its Data Linkage Program, the National Center for Health Statistics (NCHS) has successfully completed linkages of NCHS survey data to health-related administrative data sources. More information on this program can be found here: https://www.cdc.gov/nchs/data-linkage/index.htm. The resulting linked files expand upon the analytic utility of both the survey and administrative files, enabling researchers to answer complex questions and perform additional analyses that would not be possible when working with the files individually.

Recently, NCHS linked the 2016 National Hospital Care Survey (NHCS), which consists of patient- level inpatient (IP) and emergency department (ED) encounter records captured in UB-04 hospital claims and electronic health records (EHRs) provided by 158 hospitals, to CMS administrative data [4]. The 2016 NHCS linkage with the CMS Medicare Enrollment Database (EDB) involved matching 5.6 million (10^6) NHCS records and 84.6 million CMS Medicare enrollment records. Larger file sizes have a direct impact on the methods that can be used to join two files, due to the time and the computational memory needed to process and store the files.

NCHS conducted a case study exploring how to incorporate ML techniques using the SCA to improve upon traditional ad hoc joining methods. Blocking is a commonly used technique in record linkage to join records between two data sources creating potential links. Records to be joined are grouped into blocks based on specified grouping values that agree. For example, records representing individuals can be grouped into blocks based on personally identifiable information (PII) (e.g. first name, day of birth, year of birth, and state of residence, etc.) [5]. Each blocking pass consists of one or more variables that define blocks, which together are called the blocking key. The goal of blocking is to develop a set of blocking passes that minimize processing time without compromising the completeness of developed links. Efficient join criteria should minimize the number of pairs that need to be evaluated to identify most of the true matches [6].

This paper will describe the SCA ML process in detail. The paper will conclude describing the case study and show how the SCA methods improved the efficiency of joining the data sources being linked, compared to an ad hoc blocking method.

[1]Supervised ML algorithms rely on a supervisory source that contains example records which are used by the algorithm during the learning process to infer output.

2. Methods

2.1. Joining data to produce potential links for probabilistic linkage

In order to determine linkage status using probabilistic linkage methods, a set of potential links must be identified by joining records from the source data files. A Cartesian product, which generates all pairs between the files being linked, is the simplest join strategy. A Cartesian product, however, will generally be too large to be processed in a reasonable amount of time with given resources. Using the 2016 NCHS linkage to the CMS Medicare EDB as an example, performing a Cartesian product on 5.6 million 2016 NHCS records and 84.6 million CMS Medicare EDB would have resulted in approximately 475.8 trillion (10^{12}) potential linkage candidates. Due to computational limitations, generating and processing 475.8 trillion records was not a feasible option. Typically, techniques other than a Cartesian product are used when attempting to efficiently join two or more large data sources, the most common of which is blocking [6].

2.1.1. Blocking to join together two data sources—Developing a blocking strategy can be an inexact and challenging ad-hoc process. When correctly implemented, efficient blocking retains a smaller subset of potential links from the full cross-product comparison space but still includes the large majority of true matches. However, depending on the blocking strategy used, error may be introduced by removing true matches from being linked. Further, if the number of potential links is large, their evaluation can be quite resource-intensive.

2.1.2. Alternative blocking development method: Sequential Coverage Algorithm (SCA)—The SCA is a supervised ML algorithm designed to learn a set of efficient blocking keys that can used to form a blocking scheme, using both numeric and character variables. The SCA uses training data (i.e. the data sources that are being linked) and a validation dataset, also referred to as a truth deck, acting as the supervisory component. The truth deck is the subset of true matches, identified using deterministic linkage methods, from the full comparison space (i.e. Cartesian product of the NHCS and CMS datasets). A detailed description of the linkage method used is explained in Appendix I of “The Linkage of the 2016 National Hospital Care Survey to the 2016/2017 Centers for Medicare & Medicaid Services Medicare Enrollment, Claims/Encounters and Assessment Data: Matching Methodology and Analytic Considerations” [4]. A deterministic linkage was used to define the truth deck to avoid the time-intensive nature of a manual review and possible reviewer subjectivity of determining matches. While the truth deck does not necessarily need to rely on deterministic matches, prior research suggests that links from the deterministically matched truth deck may be similar to those not in the truth deck [7]. The goal of the SCA is to learn blocking keys that ‘balance’ the trade-off between the total number of records (i.e. potential links) that need to be evaluated in subsequent linkage processing and the number of captured truth deck records. In our case study, we use a modified version of the SCA described by Michelson and Knoblock [3]. Table 1 provides a brief listing of the additions to the SCA and the reason for their addition. These additions are described in greater detail throughout Section 2.1.2.1.

2.1.2.1 SCA Routine: The SCA is an iterative algorithm, learning new blocking keys with each completed iteration. As previously mentioned, the blocking keys being learned by the SCA 'balance' the trade-off between the total number of potential links and the number of true matches (from the truth deck) captured by each key. The SCA achieves this balance through processes that measure the number of pairs reduced (reduction ratio) and the number of truth deck records captured (percent complete) by each blocking key [3].

The reduction ratio (RR) measures the magnitude by which the number of pairs in the comparison space has been reduced by the learned blocking key and is defined as,

$$RR = 1 - \frac{n_b}{N_c} \tag{1}$$

where n_b are the number of records identified by the blocking key and N_c are the number of records in the full cross product comparison space. The algorithm first described by Michelson and Knoblock [3] does not impose any restriction on the size (i.e. number of potential links) that are generated by a blocking key, and the method has the potential to generate overly large sets of potential links. To prevent this, we designed an approach to specify the maximum allowed number of generated pairs from each pass. This imposes a minimum allowed RR for a blocking key to be learned. For example, if the full comparison space $N_C = 475$ trillion and the user requests that a blocking key cannot generate more than 50 million potential links, the final RR of the blocking key must be at least

$$0.9999998947 = \left(1 - \frac{50\left(10^6\right)}{475\left(10^{12}\right)}\right).$$

The percent complete (PC) measures the number of true matches that are captured by the learned blocking key and is defined as,

$$PC = \frac{t_b}{T_c} \tag{2}$$

where t_b are the number of true matches captured by the current blocking key and T_c are the total number of true matches in the truth deck Michelson and Knoblock [3] do not specify a stopping point for the SCA in terms of the total number of truth deck records captured by the SCA learned blocking scheme. Therefore, the SCA will continue learning blocking keys until all of the records in the truth deck are captured. In practice, after the majority of truth deck records have been covered by previously learned blocking keys, larger blocking passes are required to capture the few remaining truth deck records. To avoid this, the professional specifies the total proportion of coverage (TPC) that ends the development new blocking passes. The TPC measures the total proportion of truth deck records captured by all of the learned blocking keys. The TPC is computed by totaling the number of unique truth deck records captured across all blocking keys in the blocking scheme and dividing by the total number of records in the truth deck and is defined as,

$$TPC = \frac{\sum_{i=1}^{b} t_i'}{T_c} \quad (3)$$

Where t_i' represent the set of unique truth deck records (i.e. not identified by any other blocking key) captured by the blocking key.

The RR and the PC dictate the decisions made by the SCA on which block variables to add to the blocking key. Further, the newly introduced maximum block size and TPC inform the SCA whether the blocking key is expected to generate too many records and should attempt to add additional block variables to increase the RR and when to stop adding additional blocking keys to the scheme, respectively. The professional using the SCA specifies the TPC, PC, and maximum allowed block size (i.e. number of potential links) that the SCA should use when learning blocking keys.

The SCA is structured into three sequential steps: (1) Learn Block, (2) Remove Covered Pairs, and (3) Optimize. The algorithm does not proceed to the next step until the previous step has completed. With each completed iteration of the three steps, the SCA will learn a new efficient blocking key to be retained in the blocking scheme.

2.1.2.2 Learn Block step: The Learn Block step methodically builds a blocking key, using the available block variables, that satisfies the RR and PC conditions. During this step, block variables continue to be added to the key as long as they improve upon the RR and do not lower the PC below the set minimum value: at each point, the variable added is the one which is determined to bring the greatest increase in the RR. Of note, the starting PC threshold can vary and is assigned by the professional. Higher PC threshold values will result in looser blocking keys, which generate a large number of potential links. Lower PC threshold values on the other hand, result in strict blocking keys as the proportion of true matches needed in each key is low. Michelson and Knoblock's research recommends setting the initial PC threshold value at 0.5, meaning the blocking key will capture at least 50% of the available truth deck records [3].

In the approach described by Michelson and Knoblock [3], the SCA may learn a blocking key that would generate a prohibitively large number of potential links. To prevent this, the initial PC threshold can be lowered by a reduction factor of 0.05 (specified by the professional), which allows additional variables to be added to the blocking key thus reducing the number of potential links being generated. At the end of the Learn Block step, SCA verifies (1) that the blocking key has at least three block variables and (2) generates less than the specified maximum number of potential links. If both of those conditions are met, the SCA will continue to the next step, Remove Covered Pairs. Otherwise, the PC threshold is further reduced again by 0.05 and the SCA continues with the Learn Block step, adding as many block variables as possible that meet the newly reduced PC threshold. When SCA is unable to add any additional block variables, it will begin to exit the Learn Block step and re-evaluate the previously mentioned conditions. This process continues to iterate

within the Learn Block step until the expected number of generated potential pairs no longer exceeds the maximum allowed or the PC reaches a set minimum of 0.05.

In the linkage of the 2016 NHCS to the CMS EDB, 0.05 (5%) subtractions to the PC (e.g. from 0.5 (50%) to 0.45 (45%)) were sufficient to be able to add new block variables that kept the number of pairs generated to 50 million when it would have been 400 million otherwise. Further, the PC was allowed to be lowered to 0.05 (5%) (i.e. the blocking key captured at least 5% of remaining truth deck records).

2.1.2.3 Remove Covered Pairs step: Following the learn block step, the SCA proceeds to the Remove Covered Pairs step. This removes the true matches captured by the previously learned blocking key from the truth deck used to develop the next blocking pass. Without their removal, the SCA will develop keys that are redundantly targeted to true matches that have been captured by previously learned keys. Recall, the specified PC dictates the amount of truth deck records that are captured by each learned blocking key. For example, if the PC is set to 0.5, then each of the learned blocking keys will capture at least 50% of the non-captured truth deck records. The first blocking key learned by the Learn Block step will capture at least 50% of the truth deck records. Next, the SCA will remove the truth deck records captured by this learned blocking key, leaving at most 50% of the truth deck. The next blocking key to be learned by the SCA will then cover at least 50% of the remaining non-captured truth deck records, representing 25% of all records in the truth deck. Each subsequently learned blocking key continues to halve the non-captured truth deck records (i.e. blocking key three will capture 12.5% of the truth deck records, blocking key four captures 6.25%, etc.). This process continues until very few non-captured records remain. Figure 1 (below) illustrates how the records in the truth deck are captured with the learning of each new blocking key.

2.1.2.4 Optimize step: Finally, the Optimize step removes learned blocking keys that are fully contained within a newly learned key. This final step prevents redundant blocking keys (i.e. the entire set of potential links from one blocking key are also fully contained within another blocking key) from being retained in the blocking scheme. For example, if SCA previously learned *Blocking Key 1 = {First name, Last name, Year birth, State residence}* and in a subsequent iteration learned *Blocking Key 2 = {First name, Last name, Year birth}*, the previous blocking key (blocking key 1) is fully contained in the new less restrictive key (blocking key 2) and blocking key 1 is removed as a blocking key.

The SCA sequentially iterates through each of the three steps (Learn Block, Remove Covered Pairs, and Optimize), adding and removing redundant learned blocking keys to and from the blocking scheme with each completed iteration. At the end of each iteration of the three steps, the SCA computes the TPC across all learned blocking keys, and then compares the result to the specified TPC target. The SCA continues to add blocking keys to the blocking scheme until either no new blocking keys can be learned from the remaining non-captured true matches or the total captured true matches across all learned blocking keys reaches the TPC target. Of note, the total number of potential links generated from a learned blocking scheme increases as a consequence of a higher specified TPC. As the TPC increases, additional blocking keys are generated to capture the additional truth deck records

(see Fig. 1 above). Figure 2 illustrates how the specified TPC affects the total number of potential links generated by the SCA learned blocking scheme.

Using Fig. 2 as an example, in order to capture 99% of the truth deck records, the SCA may learn 6 blocking keys which generate an estimated 45 million potential links. However, in order to capture 99.7% of the truth deck records, the SCA may learn 8 blocking keys which generate 100 million potential links. As a result of this 0.7% increase in the coverage of the truth deck the SCA learned 2 additional blocking keys that added an additional 55 million potential links that need to be evaluated. Professionals must decide which level of coverage is appropriate for the linkage they are conducting.

3. Case study

The SCA methodology described above was used in a recent linkage of NCHS survey data to CMS administrative data. The sources used in the linkage are described below. The SCA is compared to ad-hoc traditional methods used for blocking. The ad-hoc blocking methods used to define a join strategy included a single blocking key which retained record pairings that agreed on the last 4-digits of Social Security Number (SSN) or Health Insurance Claim Number (HICN), month of birth, day of birth, and sex. SSN is a unique identifier assigned to individuals by the Social Security Administration [8]. HICN, is a unique identifier assigned by CMS to each beneficiary enrolled in the Medicare health insurance program.

3.1. Data sources

3.1.1. National Hospital Care Survey Description—The NHCS is one of the NCHS National Healthcare Surveys, a family of surveys covering a wide spectrum of healthcare delivery settings from ambulatory and outpatient department (OPD) to hospital and long-term care providers. The NHCS is an establishment survey that collects patient encounter records for all IP, ED, and OPD episodes of care from participating surveyed hospitals. The goal of the NHCS is to provide reliable healthcare utilization data from hospital-based settings and is designed to be used by healthcare professionals, researchers, and policy makers seeking answers to crucial healthcare related questions. More detailed information regarding the types of hospitals, number of participating hospitals, and completeness of the PII collected in the 2016 NHCS is described elsewhere [4].

Although the NHCS is an establishment survey (i.e. the sampling frame are hospitals), patient PII are available on the reported hospital encounters which enables linkage of the NHCS with other healthcare related data sources such as CMS' Medicare research data files. Patient PII includes SSN, HICN, full name (first, middle, and last), date of birth (month, day, and year), sex, 5-digit ZIP code and state of residence. Of note, prior to starting the linkage process of the 2016 NHCS to the CMS data, all PII variables from the NHCS were assessed for their linkage quality and only those patients with sufficient PII were eligible for linkage [4]. The analysis in this paper refers to the linkage of IP and ED encounter data for eligible patients from the 2016 NHCS to the CMS Medicare EDB.

3.1.2. CMS Medicare EDB—The CMS Medicare EDB is an administrative data system that stores PII for all eligible beneficiaries enrolled in the Medicare program. PII that are

stored in this system include, SSN, HICN, full name (first, middle initial, last), date of birth (month, day, and year), sex, 5-digit ZIP code and state of residence. Prior to starting the linkage process of the 2016 NHCS to the CMS data, CMS Medicare data were assessed for data quality and there were no issues to note in the variables used for blocking. Further, enrollment records for Medicare beneficiaries who expired before 2016 or who were born after 2016 were excluded. After the completion of the patient record linkage, corresponding 2016–2017 CMS Medicare health care utilization data were extracted for the links made from the CMS Medicare EDB. These linked data files are available, upon request, to researchers through the NCHS and Federal Statistical Research Data Centers. Of note, patient PII are only used for linkage purposes and have been removed from all linked data files that are made available to researchers.

3.1.3. Training and supervisory data—The training data used were the 2016 NHCS and CMS Medicare EDB submission files. Prior to conducting linkage, the data values (i.e. PII) in each source were first subject to a cleaning routine that removed any invalid data values [4]. For example, SSN values that contained fewer than 9 digits were changed to a null value. Following this initial data pre-processing, the PII fields were combined into submission records for each data source. The truth deck that was used in this exercise (i.e. the linkage of the 2016 NHCS to CMS Medicare EDB) was the subset of true matches, from the full comparison space, identified using deterministic linkage methods. NHCS and CMS records that matched on exact SSN or HICN and match with a majority of non-missing other PII variables (e.g. name, date of birth) also in agreement were retained into the truth deck [4]. Ideally, the training and the truth deck (supervisory data) should be of high quality and diverse enough to cover as many true matches as possible [6]. Of note, available literature on the SCA does not specify the number of minimum records that are needed in the training data or truth deck. However, because the SCA learns using the records in the truth deck as training examples, a truth deck that contains very few records will likely result in poor learning. Therefore, the truth deck should contain as many examples (i.e. records pairs) as the data allow.

4. Results of case study

4.1. Ad hoc blocking methods

In the linkage of the 2016 NHCS to the CMS Medicare EDB there were 1,598,511 records in the truth deck that were used to compare the results of two blocking strategy design approaches (traditional ad hoc methods and SCA). The first method relied on an ad hoc method of specifying blocking passes (i.e., the blocking keys associated with them) based on professional judgement. Table 2 provides information on the number of potential links returned by the ad hoc blocking rule (agreement on the last 4-digits of SSN or HICN, month of birth, day of birth, and sex), and the percentage of the truth deck that was captured by the rule.

Table 2 shows that the single blocking key generates approximately 11.4 million potential links and captures 96.8% of the truth deck matched pairs. Of note, because the key requires that each of the paired records have non-missing values for the variables compared, records

that have missing values for any of the blocking key values will not be included in generated pairs. For example, 71.9% of the patients in the 2016 NHCS did not have an SSN or HICN, and therefore would be excluded from any further linkage analysis using the ad-hoc key, even as they may very well have a correct match from the EDB.

Because of the potential shortcoming with the ad hoc methods noted previously, NCHS explored the use of SCA. Results for the SCA method are provided in Section 4.2 (below). The section provides the total number of potential links and truth deck coverage for the SCA learned blocking scheme.

4.2. SCA performance

In the production run of the 2016 NHCS linkage to CMS Medicare EDB, the SCA used a 33% sample of the 2016 NHCS and CMS Medicare EDB linkage submission files (described in Section 3.2) as the training data and their corresponding deterministic matches from the truth deck as the supervisory source used to develop the blocking schemes described below. Further, a starting PC threshold value of 0.5 and requested a TPC of 99.8% (i.e. the blocking scheme should capture at least 99.8% of records in the truth deck). Lastly, we specified a maximum block size of 50 million pairs.

The SCA version used in the production run of the 2016 NHCS linkage to the CMS Medicare EDB produced a blocking scheme of 6 blocking keys. Table 3 lists the 6 blocking keys, the number of generated potential links, and the number of truth deck records captured by each.

The 6 blocking keys learned by the SCA generated a total of 101.8 million potential links that would later be processed by probabilistic linkage methods. Note that the 101.8 million potential links (which include duplicated pairs) include records without SSN and HICN, which would have been excluded by the ad-hoc blocking scheme described above. Because the 6 blocking keys are not mutually exclusive, i.e. potential links identified in one blocking key may be identified in others, the TPC cannot be computed from Table 3. Instead it was computed by identifying the set of unique truth deck records across all 6 blocking keys and dividing by the total number of records in our truth deck. The 6 blocks captured 1,578,912 (98.8%) of the total 1,598,511 records in our truth deck. The SCA was unable to capture the additional 1% of truth deck records needed to satisfy the requested 99.8% TPC without creating a substantially large blocking key (greater than the 50 million maximum) so the algorithm stopped before reaching that coverage. This was caused by matches in the truth deck with fewer matching block variables.

5. Conclusion

The SCA, a supervised ML algorithm, was used to join the 2016 NHCS and CMS Medicare EDB data sources. While both the ad-hoc and SCA methods had good coverage of the truth deck, the SCA captured an additional 2% of the truth deck records and allowed for additional records to be included in the linkage process which were dropped from further evaluation in the ad hoc approach. Further, the SCA produced a blocking scheme that efficiently joined the survey and administrative data sources while maintaining a majority

of the records from the truth deck. However, it should be noted that like all supervised ML algorithms, the SCA is sensitive to the quality of training and supervisory data being used to learn the blocking keys. Low-quality data with severe data integrity issues (i.e. data transpositions or a high proportion of missingness), may result in a suboptimal generated blocking strategy. Additional research to explore the impact of PII data quality and the size of the training data set should provide users with additional guidance on how to apply SCA techniques (i.e., what are optimal values for user defined settings).

In this case study, the SCA allowed the professional to specify a maximum block size, thereby avoiding learning less efficient blocking keys (i.e. larger number of generated potential links) at the expense of a few additional iterations of the Learn Block step. In the production run of the 2016 NHCS linkage to the CMS Medicare EDB, the SCA learned an efficient blocking scheme consisting of 6 blocking keys which generated 101.8 million potential links and captured 98.8% of true matches in our truth deck. This ML methodology is being incorporated into other linkages in the NCHS Data Linkage Program.

Funding

This work was supported in part with funding from the Department of Health and Human Services' Office of the Secretary Patient Centered Outcomes Research Trust Fund (OS-PCORTF)

References

[1]. Kelleher J, Tierney B. Data Science, The MIT Press, 2018.

[2]. Giang PH. A machine learning approach to create blocking criteria for record linkage. Health Care Manag Sci20153; 18(1): 93–105. doi: 10.1007/s10729-014-9276-0.Epub 2014 Apr 29. PMID: 24777833. [PubMed: 24777833]

[3]. Michelson M, Knoblock CA. Learning Blocking Schemes for Record Linkage, In: AAAI. Boston, 2006.

[4]. National Center for Health Statistics. Division of Analysis and Epidemiology. The Linkage of the 2016 National Hospital Care Survey to the 2016/2017 Centers for Medicare & Medicaid Services Medicare Enrollment, Claims/Encounters and Assessment Data: Matching Methodology and Analytic Considerations, 82020. Hyattsville, Maryland. https://www.cdc.gov/nchs/data/datalinkage/NHCS-CMS-Medicare-Llinkage-Methods-and-Analytic-Considerations.pdf.

[5]. U.S. Department of Labor, Guidance on the Protection of Personal Identifiable Informationhttps://www.dol.gov/general/ppii#:~:text=Personal%20Identifiable%20Information%20(PII)%20is,either%20direct%20or%20indirect%20means.

[6]. Christen P. Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection, Springer, 2012.

[7]. National Center for Health Statistics. Vital and Health Statistics. Assessing Linkage Eligibility Bias in the National Health Interview Survey. 32021. Hyattsville, Maryland. Vital and Health Statistics, Series 2, Number 186https://www.cdc.gov/nchs/data/series/sr_02/s02-186-508.pdf.

[8]. Social Security Administration. Office of Retirement and Disability Policy. The Story of the Social Security Number. Social Security Bulletin, Vol. 69, No. 2, 2009. https://www.ssa.gov/policy/docs/ssb/v69n2/v69n2p55.html.

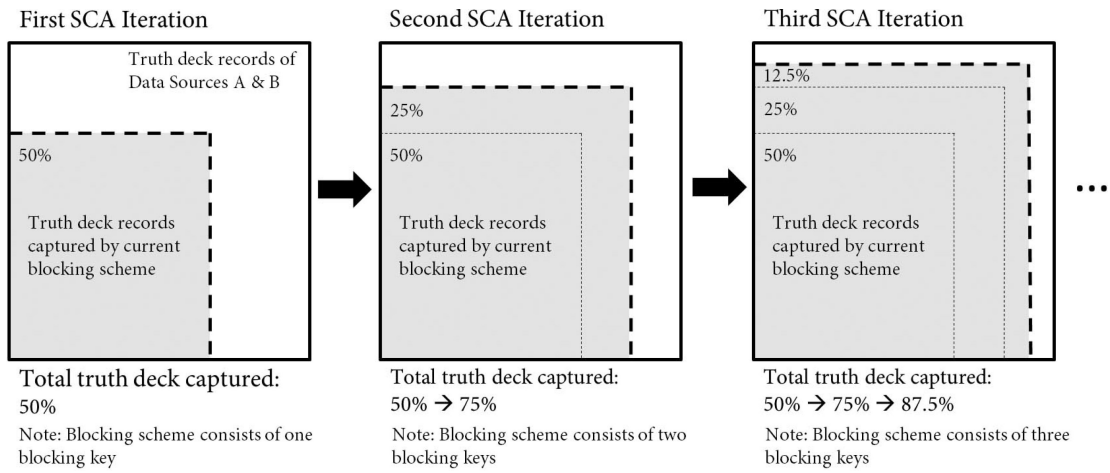
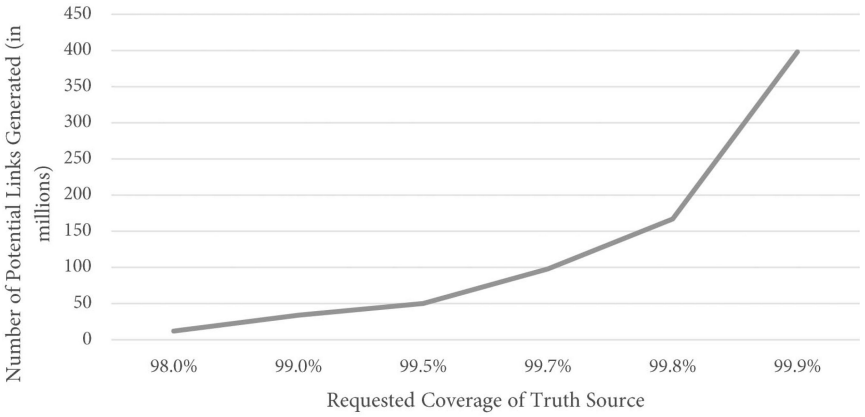


Fig. 1.

Theoretical example of truth deck records captured by SCA learned blocking scheme.



Note: Not based on actual data

Fig. 2.

Performance of SCA learned blocking schemes at different requested truth source coverage values.

Table 1

Modifications made to Sequential Coverage Algorithm (SCA)

Addition	Reason for addition
Restrict number of potential links generated from a single blocking pass	To avoid prohibitively large (i.e. number of potential link) blocks
Percent complete (PC) can be reduced during learn block step	Allows additional block variables to be added to the block key thus reducing number of potential links (see previous entry)
Specification of Total percent covered (TPC)	Provides a stopping point to end development of new blocking passes

Stat J IAOS. Author manuscript; available in PMC 2021 August 18.

Table 2

Evaluation of efficiency of ad-hoc blocking key previously used by the National Center for Health Statistics (NCHS)

Blocking key	Total cross product (in trillions)	Potential links to be evaluated	Total matches from truth deck	Captured truth deck records
Last 4-digits of SSN or HICN, month birth, day birth, sex	475.8	11,381,076	1,598,511	1,546,710 (96.8%)

Note: Potential links generated using 2016 National Hospital Care Survey (NHCS) and the Centers for Medicare and Medicaid Services (CMS) Medicare Enrollment Database (EDB). Note: Social Security Number (SSN) is a unique identifier assigned by the Social Security Administration (SSA). Note: Health Insurance Claim Number (HICN) is a unique identifier assigned by CMS.

Stat J IAOS. Author manuscript; available in PMC 2021 August 18.

Table 3

Sequential Coverage Algorithm (SCA) learned blocking scheme in the linkage of the 2016 NHCS to the CMS Medicare EDB

Blocking key	Number of block variables	Number of potential links	Number of truth deck records	Percent of truth deck	Cumulative percent of truth deck coverage
Day of birth, month birth, year birth, ZIP code residence	4	2,288,489	1,167,469	73.0	73.0
First name, last name, month birth, year birth	4	2,314,896	1,137,532	71.1	91.6
First name, day birth, month birth, year birth, sex	5	46,436,412	1,236,422	77.3	94.1
Last name, day birth, month birth, year birth, sex	5	4,538,784	1,363,757	85.3	97.9
First name, last name, state residence	3	39,035,154	1,071,595	67.0	98.6
Middle initial, day birth, month birth, year birth, state residence, sex	6	7,235,868	725,609	45.4	98.8

Note: Potential links generated using 2016 National Hospital Care Survey (NHCS) and the Centers for Medicare and Medicaid Services (CMS) Medicare Enrollment Database (EDB).