Blocking records for probabilistic record linkage using approximate nearest neighbours algorithms

Maciej Beręsewicz
Department of Statistics, Poznań University of Economics and Business
Centre for the Methodology of Population Studies, Statistical Office in Poznań

 BERENZ / ncn-foreigners / ojalab
 / 𝕏 @mberesewicz
mberesewicz.bsky.social

uRos 2024

POZNAŃ UNIVERSITY
OF ECONOMICS
AND BUSINESS

Contents

Contents

Motivation

- Official statistics is using data from various sources from different entities (e.g. central or local governments; private entities)
- Data integration is needed to use these sources in a more complete way.
- Official statistics requires from the entities to provide identifiers (e.g. personal identifiers, PESEL; business entity identifiers; NIP/REGON).
- However, there are missing data in identifiers or some source do not contain identifiers.
- Therefore, we need to link the data using probabilistic methods, for instance probabilistic record linkage.
- This is an ongoing work with: *Tiziana Tuotto, Laura Tosco* and *Loredana Di Consiglio* (Istat) and *Tymoteusz Strojny* (PUEB, Statistical Office in Poznań).

Motivation

Figure 1: Number of foreign-born people at the end of 2021

Register	Number of foreign born people
Population Register (PESEL)	2,004,765
Register of Insured (ZUS)	957,539
Register of tax payers (MF)	1,513,129
Agricultural Social Insurance Fund (KRUS)	67,932
National Health Fund (NFZ)	2,034,434
Register of foreigners (various documents on stay)	54,5873

Motivation

Table 1: Stages of data cleaning of registers

Register	Stage 1	Stage 2	No identifiers
KRUS	4,674	18,317	42,693
MF	1,043,769	1,132,840	352,088
NFZ	1,987,884	1,987,891	42,524
ZUS	624,113	760,765	117,926
All	1,988,650	1,989,390	–

Contents

Probabilistic record linkage

A THEORY FOR RECORD LINKAGE

Ivan P. Fellegi and Alan B. Sunter
Dominion Bureau of Statistics

A mathematical model is developed to provide a theoretical framework for a computer-oriented solution to the problem of recognizing those records in two files which represent identical persons, objects or events (said to be *matched*).

A comparison is to be made between the recorded characteristics and values in two records (one from each file) and a decision made as to whether or not the members of the comparison-pair represent the same person or event, or whether there is insufficient evidence to justify either of these decisions at stipulated levels of error. These three decisions are referred to as *link* (A_1), a *non-link* (A_3), and a *possible link* (A_2). The

Probabilistic record linkage

COMPUTER SCIENCE

(Almost) all of entity resolution

Olivier Binette[1] and Rebecca C. Steorts[2,3]*

Whether the goal is to estimate the number of people that live in a congressional district, to estimate the number of individuals that have died in an armed conflict, or to disambiguate individual authors using bibliographic data, all these applications have a common theme—integrating information from multiple sources. Before such questions can be answered, databases must be cleaned and integrated in a systematic and accurate way, commonly known as structured entity resolution (record linkage or deduplication). Here, we review motivational applications and seminal papers that have led to the growth of this area. We review modern probabilistic and Bayesian methods in statistics, computer science, machine learning, database management, economics, political science, and other disciplines that are used throughout industry and academia in applications such as human rights, official statistics, medicine, and citation networks, among others. Last, we discuss current research topics of practical importance.

Contents

Basic idea of blocking

- The goal of the blocking is to reduce number of comparisons by fixing certain variables (e.g. age, sex, first letter of surname, post-code).
- It is assumed that blocking variables are measured without error and no missing data).
- However, what about linking data where none of variables can be used for blocking due to missing data.

Problems with the data

Table 2: Example rows from Polish registers

First name	Second name	Surname	Birth year
Miguel	Luis	Pereira Tinoco	1969
Miguel Luis		Pereira-Tinoco	1969
Miguel		Pereira-Tinoco	1968
Thi	Hy	Dao	1993
Dao Thi Hy			1993
Thi		Dao	1992

Contents

Approximate Nearest Neighbours Algorithms

Nearest neighbor search (NNS), as a form of proximity search, is the optimization problem of finding the point in a given set that is closest (or most similar) to a given point. Closeness is typically expressed in terms of a dissimilarity function: the less similar the objects, the larger the function values (Wikipedia 2024).

- Goal: trade-off between accuracy and speed.
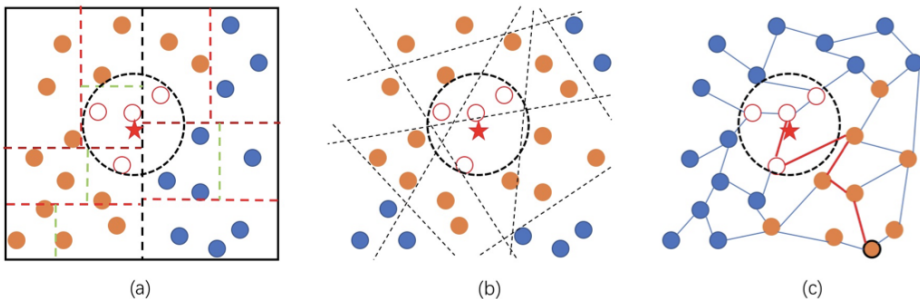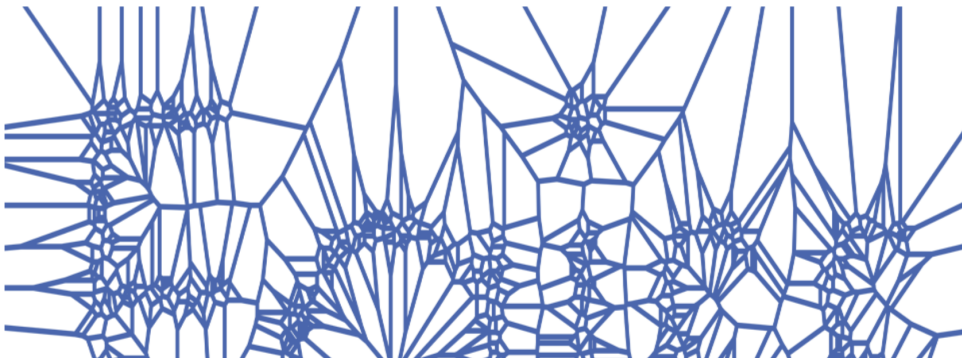
Idea behind the ANN algorithms

Figure 2: Types of algorithms: a) box, 2) hyperplanes, 3) graphs. Source: Fu, C., Xiang, C., Wang, C., & Cai, D. (2017). Fast approximate nearest neighbor search with the navigating spreading-out graph
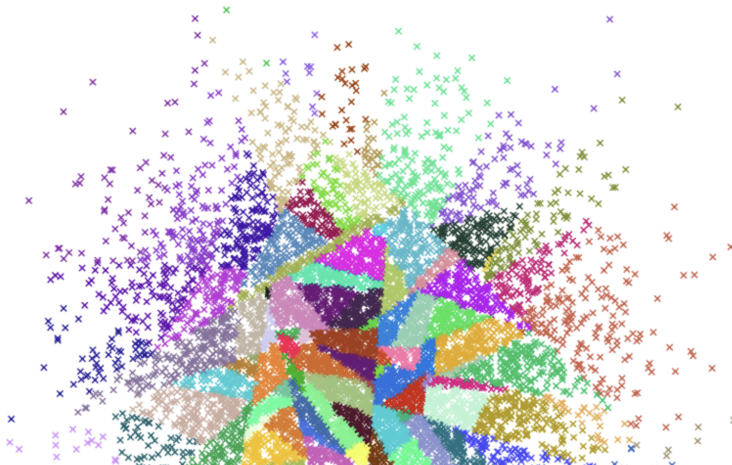
Where we can find the ANN algorithms?

Faiss: A library for efficient similarity search

Where we can find the ANN algorithms?

Annoy

Where we can find the ANN algorithms?

README   Code of conduct   Apache-2.0 license

VOYAGER

License Apache 2.0   Documentation on github.io   platforms macOS | Windows | Linux   Apple Silicon macOS and Linux
Compile, Test, and Deploy passing

Voyager is a library for performing fast approximate nearest-neighbor searches on an in-memory collection of vectors.

Voyager features bindings to both Python and Java, with feature parity and index compatibility between both languages. It uses the HNSW algorithm, based on the open-source `hnswlib` package, with numerous features added for convenience and speed. Voyager is used extensively in production at Spotify, and is queried hundreds of millions of times per day to power numerous user-facing features.

Where we can find the ANN algorithms?

rnndescent

An R package for finding approximate nearest neighbors, translated from the Python package PyNNDescent written by the great Leland McInnes. As the name suggests, it uses the Nearest Neighbor Descent method (Dong et al., 2011), but also makes use of Random Partition Trees (Dasgupta and Freund, 2008) as well as ideas from FANNG and NGT.

You can use rnndescent for:

- optimizing an initial set of nearest neighbors, e.g. those generated by RcppAnnoy or Rcpp-HNSW.
- using this package for nearest neighbor search all on its own...
- ... including finding nearest neighbors on sparse data, which most other packages in the R ecosystem cannot do.
- and a much larger number of metrics than most other packages.

The procedure

1. Process text data to remove spaces, commas itp
2. Create q-grams and store data in sparse matrix
   - {ma, ac, ci, ie, ej}
   - {mac, aci, cie, iej}
3. Create an index using a specified algorithm
4. Query the index with the same (deduplication) or new data (record linkage)
5. Create a graph to cluster units
6. **Clusters – blocks for deduplication/record linkage**

Contents

Blocking

blocking 0.1.0   Reference   Articles ▾   Changelog

Overview

Links

[Browse source code](#)

[Report a bug](#)

License

[GPL-3](#)

Citation

[Citing blocking](#)

Developers

Maciej Beręsewicz
Author, maintainer ⬤

Dev status

R-CMD-check failing

test-coverage passing

Warning!

The package is still being developed, so the API and features may change.

Description

This R package is designed to block records for data deduplication and record linkage (also known as entity resolution) using [approximate nearest neighbours algorithms (ANN)](#) and graphs (via the `igraph` package).

It supports the following R packages that bind to specific ANN algorithms:

- [rnndescent](#) (default, very powerful, supports sparse matrices),
- [RcppHNSW](#) (powerful but does not support sparse matrices),
- [RcppAnnoy](#),

Blocking

Block records based on text data.

Source: R/blocking.R

Function creates shingles (strings with 2 characters, default), applies approximate nearest neighbour (ANN) algorithms via the rnndescent, RcppHNSW, RcppAnnoy and mlpack packages, and creates blocks using graphs via igraph.

Usage

```
blocking(
  x,
  y = NULL,
  deduplication = TRUE,
  on = NULL,
  on_blocking = NULL,
  ann = c("nnd", "hnsw", "annoy", "lsh", "kd"),
  distance = c("cosine", "euclidean", "l2", "ip", "manhatan", "hamming", "angular
  ann_write = NULL,
  ann_colnames = NULL,
  true_blocks = NULL,
```

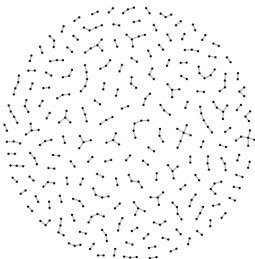# Blocking

```
str(df_blocks,1)                                                              
#> List of 7
#>  $ result      :Classes 'data.table' and 'data.frame':  255 obs. of  4 variab
#>   ..- attr(*, ".internal.selfref")=<externalptr>
#>  $ method      : chr "nnd"
#>  $ deduplication: logi TRUE
#>  $ metrics     : NULL
#>  $ confusion   : NULL
#>  $ colnames    : chr [1:429] "86" "ap" "av" "bf" ...
#>  $ graph       :Class 'igraph'  hidden list of 10
#>  - attr(*, "class")= chr "blocking"
```
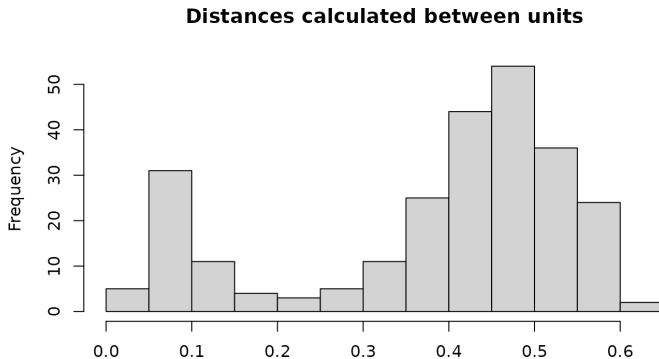
Blocking

```
plot(df_blocks$graph, vertex.size=1, vertex.label = NA)
```

Blocking

```
hist(df_blocks$result$dist, xlab = "Distances", ylab = "Frequency", breaks = "fd"
     main = "Distances calculated between units")
```

Distances calculated between units

Simulation study – setup

- Repository with codes: https://github.com/ncn-foreigners/uRos2024-blocking
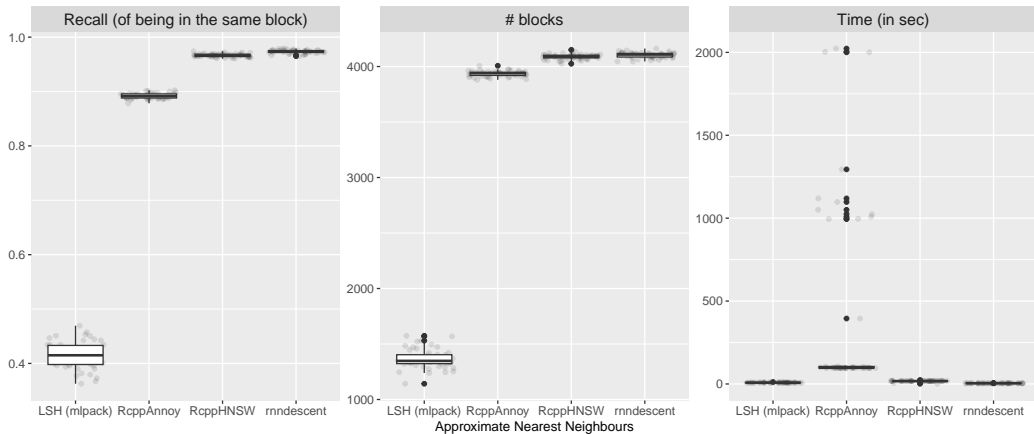- Dataset: 10,000 records
- Duplicates: 3,000 records
- Max no. duplicates: 1
- Max modifications by record: 3
- Max modifications by attribute: 3
- Include missing data
- 50 datasets created for this simulation

Simulation study

Description: dt [6,000 × 12]

rec_id <chr>	first_name <chr>	second_name <chr>	last_name <chr>	region <chr>	birth_date <chr>	personal_id <chr>
rec–0057–dup–0	MARCELINA	KATARZYNA	OLSŽEVSKA		11/02/1946	KJJ536470
rec–0057–org	MARCELINA	KATARZYNA	OLSZEWSKA	OPOLSKIE	11/02/1946	KJJ536480
rec–0059–dup–0	LENA		OLEJNYIK	MAZOWIECKIE	12/06/1976	AEY444103
rec–0059–org	LENA	KINGA	OLEJNIK	MAZOWIECKIE	12/06/1967	AEY444103
rec–0061–dup–0	HANNA	BARBARA	BUKOŁSKA		23/12/2008	
rec–0061–org	HANNA	BARBARA	BUKOWSKA	DOLNOŚLĄSKIE	23/12/2008	APH078531
rec–0062–dup–0	FATYMA	MELEK	KATZZMAREK	MAZOWIECKIE	23/03/1965	
rec–0062–org	FATIMA	MELEK	KACZMAREK	MAZOWIECKIE	23/03/1965	YHG665688
rec–0064–dup–0	MAJA		KOSTKA		14/05/1974	LRP75u524
rec–0064–org	MAJA	MARIA	KOSTKA	ZACHODNIOPOMORSKIE	14/05/1974	LRP758524

Simulation results

Summary

- Links:
  - blocking (R library) – https://ncn-foreigners.github.io/blocking/
  - blockingpy (python library) – https://github.com/T-Strojny/BlockingPy
  - geco3 (python library) – https://github.com/T-Strojny/geco3
- We have developed blocking method that allows to use variables measured with error (no ground truth).
- The method significantly reduce number of comparisons (similarly as microclustering).
- Further studies require assessment of the quality under different scenarios.
- The proposed method can be easily used with existing tools for probabilistic record linkage (e.g. reclin2 in R).
- We test the package with the tinytest package.

Summary

blocking [package] and FS [Fellegi-Sunter] model is absolutely the best performer in terms of execution time, and as good as the other in terms of precision and recall

Tiziana Tuoto and colleagues on usage of the `blocking` package for application for the maritime transports.

Thank you!