Ensemble machine learning for record linkage

Daniel Casey

Things I’ll be talking about

1. How record linkage works (I think)
2. My use case
3. Data cleaning and training
4. Ensemble modelling
5. Results
6. Some comparisons to `fastLink` and `multilink`
7. Final musings, including thoughts on child-rearing

An illustrated guide to record linkage

With words this time

1. Clean data
2. Identify variables shared between datasets that might indicate a match
3. Generate lists of possible pairs using blocking (a priori restrictions on whether any two records can be a match, more on this later)
4. Compute distance metrics for the variables specified in #3 between Person A and Person B in the pair
5. With the variables of #4 as your dataset, do some math to guess whether the A and B represent the same person

#3 - #5 is usually obscured from view by record linkage interfaces (e.g. `fastLink` or `multilink`).

Review `?stringdist::stringdist` for all the lovely permutations various scholars have devised

Linking CHS EPIC data and WIC records (the point)

- Community Health Services (CHS) provides medical and support services to residents of King County. This client base is maintained/logged via EPIC
- CHS also administers Women Infant and Children (WIC) program on behalf of the state/Feds in King County, but in the Cascades data system
- It is difficult to tell how many clients served by CHS are also on WIC because the data-systems don’t like each other
- The following machine learning approach was designed to link WIC data with CHS EPIC data
- Potential matches are mostly children under 5 years old and their women/mothers (WIC client criteria)

Mission Really Quite Possible

Cleaning data (ugh)

- Names (first, last, and middle initial): YOU GET AN UPPER CASE AND YOU GET AN UPPERCASE. YOUALSOLOSEALLNONLETTERCHARACTERS
  - Also, if your name is TRUE or 1, your name gets changed to True
- ZIP code: first five numbers
- dob: check to see if swapping month and day results in a legitimate date. Also, make sure the column is literally a date.
- Make telephone numbers all numeric
- Send addresses through the `kcgeocode` address cleaning and geocoding process

It’s dangerous to go alone! Take this.

I am writing a R package called `hyrule` to help with data cleaning and other tasks related to ML record linkage.

The main cleaning function, `hyrule::prep_data_for_linkage` cleans relatively standard administration data fields (e.g. name, dob, and ZIP).

Blocking

- Blocking is the a-priori exclusion of possible pairings
- It is possible to block on all sorts of things: age, gender, SSN, favorite conspiracy theory, etc. if you have the data for it.
- You can do multiple rounds of blocking, but then you probably have to reconcile things at the end.
- It is best to block on variables with relatively low missingness
- This process blocked on year of birth and required one of the following:
  - day of birth for Person A == day of birth for Person B
  - month of birth for Person A == month of birth for Person B
  - day of birth for Person A == month of birth for Person B
  - Either person’s birthday is Jan 1st

Boring variables, part 1

- `dob_ham`: Hamming distance between DOBs
- `fn_cos2`: Cosine bigram distance between first names.
- `fn_jw`: Jaro-winkler distance between first names
- `fn_sx`: First name soundex comparison
- `ln_cos2`: Last name cosine bigram distance
- `ln_jw`: Jaro-winkler distance between last names
- `ln_sx`: Last name soundex comparison
- `cn_cos`: Complete name (FIRSTNAMELASTNAME) trigram distance
- `daymonth`: binary flag indicating a shared day and month

These were created with `hyrule::compute_variables`

Boring variables, part 2

- `sex_disagree`: Explicit disagreement in sex between the pair
- `midinitmatch`: Explicit agreement in middle initials
- `midinitna`: One of the middle initials is missing
- `cn_cos` adjuted to be FIRSTNAMEMIDDLENAMELASTNAME trigram distance in case that is better

Generating variables, the *cool* ones

1. `zip_Mm`: minimum mega-meter distance between ZIP code centriods given a pair's address history
2. `exact_address`: whether the minimum distance between a pair's address history is less than 10 feet
3. `phone_dist`: minimum hamming distance between phone numbers
4. `nphonepeeps`: minimum number of people associated with a phone number associated with the person. This metric is computed per dataset-phone number combination and the minimum value between datasets for a given pair is used. Phone numbers with >10 people associated with it (usually garbage numbers) and NA values are imputed with the average non-NA <10 value
5. `pos_twins`: Whether someone with the same phone number has the same birthday in one of the datasets.

Making a training dataset

1. Generate possible pairs
2. Figure out some way to sort them by probability of match
    1. I used a model fit on fake data (and now you can ask me for a model fit on real data)
    2. `fastLink` or `multilink`
    3. Trigram cosine difference
3. Sample possible pairs, ideally in strata (deciles?) to create a training dataset (200 - 400 instances)
4. Load them up in `hyrule::matchmaker` and identify some matches!
    1. Some are obvious
    2. Some are vibes/maybes/an existential question

ROCKY TRAINING MONTAGE

Iterate, iterate, iterate

Some ways to make your model better:

- Add more training data by manually reviewing pairs that the machines are unsure about
- Add more variables
- Conduct test/train validation
- Compare with probabilistic linkages to identify discordance and manually review those cases

DO NOT BE AFRAID OF VIBES:

- Making training data
- Deciding what children models to use
- Match cutoff
- Manually identifying/determining matches

Results

- 166,627 records in the CHS/EPIC/clinic data were matched against 66,074 records in the WIC data. Both datasets cover Q3 2019 - Q4 2022 (and maybe a little more for the CHS data)
- **The main machine learning approach identified 38,420 pairs**

Number of matches

Machine Learning	ML Subset	multilink	fastLink
38,420	38,295	37,233	33,241

Cage Match, set up!

- Conduct similar matching using `multilink` and `fastLink`
  - Blocking by year of birth
  - First name, last name, sex, ZIP and data of birth
  - Fancy variables were omitted for being a pain
  - ML Subset is a machine learning model fit without the fancy variables

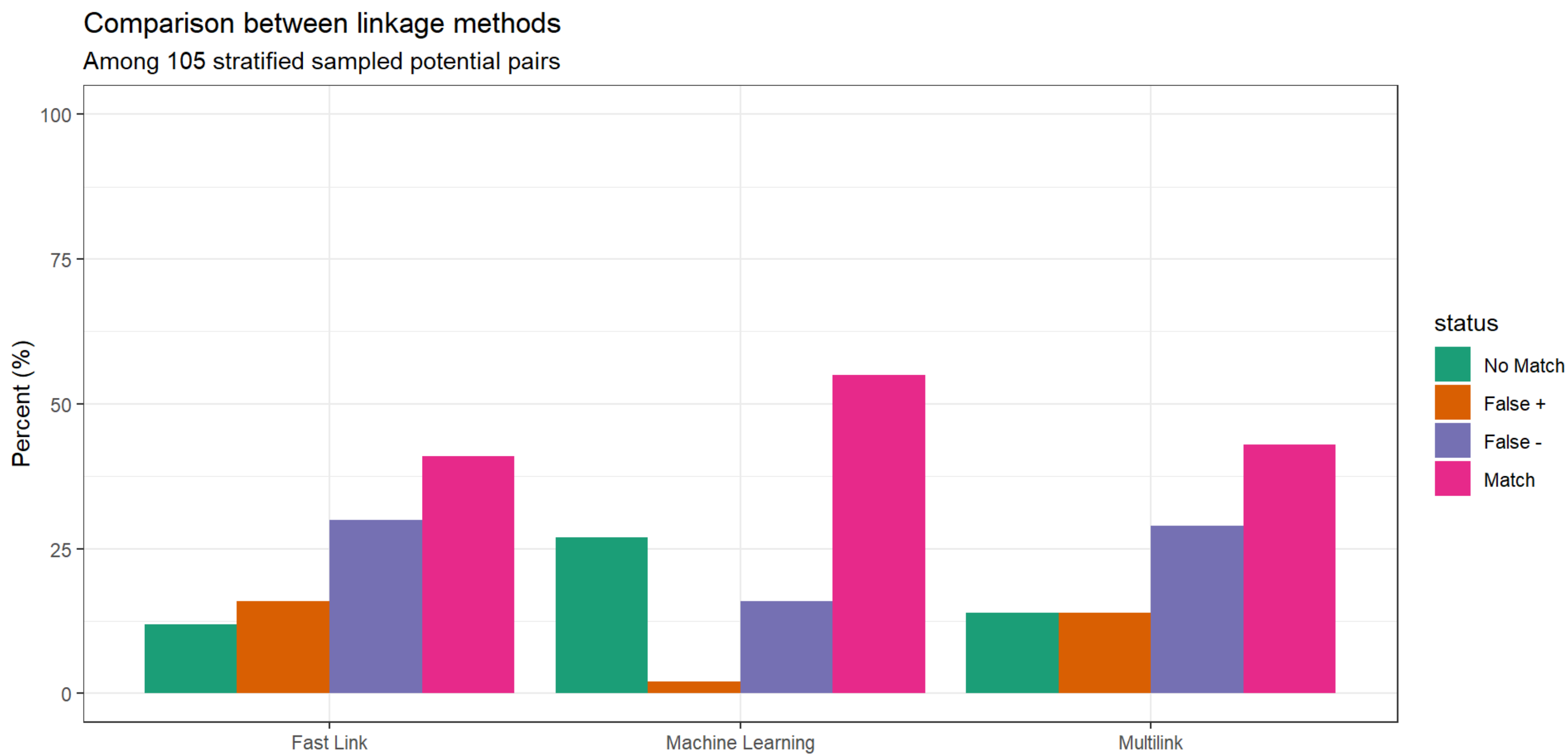
Cage Match, fight!

Match comparison

MachineLearning	multilink	fastLink	N
0	0	1	101
0	1	0	89
0	1	1	42
1	0	0	553
1	0	1	765
1	1	0	4,769
1	1	1	32,333

Cage Match, decision!

15 pairs from each of the various permutations of fastLink vs. multilink vs. machine learning (see previous table) were manually evaluated for matchy-matchyness.



Test/train

status	True Negative	False Negative	True Positive	False Positive	Correct %
test	80	9	135	0	96
train	230	33	412	0	95

Assuming Correct % is higher for test then train when you see this: That is weird/unusual (a model should be better at predicting data its already seen). I'm chalking this up to a feisty random number generator.

Some advice on children and their names

1. Don’t have multiple children at once
2. If you ignore #1, please name them different things. Consider not sharing any letters and/or varying the length of their names
3. Do not name your children (or yourself) after logical statements (e.g. True). While the name might be nice, computers hate it and computers are our overlords

Merci

1. CHS: Nathan Dye, Leif Layman, and Lee Thornhill
2. APDE: Danny Colombara, Eli Kern, Alastair Matheson, and Precious Esie
3. DOH: Sean Coffinger
4. Dearly departed (from PHSKC): Tigran Avoundijan
5. Loving audience
6. Other people I forgot