BRUEGEL WORKING PAPER 2014/10ii ii

REMERGE: REGRESSION-BASED RECORD LINKAGE WITH AN APPLICATION TO PATSTAT

MICHELE PERUZZI*, GEORG ZACHMANN** AND REINHILDE VEUGELERS†

Highlights

- Record linkage algorithms typically find matches by comparing records on the fields they share. However, PATSTAT shares very little information with company databases. We introduce REMERGE: a flexible, open-source algorithm that allows PATSTAT, the worldwide patent database, to be intelligently linked with company databases, without limiting the comparisons to the shared fields. The results of this matching application can be used to improve research into the economics of innovation. The algorithm could also be adapted for similar problems. We provide a description of our algorithm, together with details on the coverage on a by-country and by-sector basis, performance measures, and hints for future research. We also show results from an additional application of REMERGE to the European Commission's Tenders Electronic Daily database.

* Bruegel; michele.peruzzi@bruegel.org
** Bruegel; georg.zachmann@bruegel.org
† KU Leuven and Bruegel; reinhilde.veugelers@bruegel.org

This work was supported by the SIMPATIC project (EU Seventh European Union Framework Programme, grant agreement no. 290597). The python algorithm was developed on the basis of previous work carried out by Mark Huberty, Mimi Tam and Amma Serwaah at Bruegel.

SEPTEMBER 2014

Table of Contents

Introduction

The purpose of a record linkage algorithm is to identify records in different databases that refer to the same entity. Typically, records belonging to different sources are compared based on some similarity measures on the shared fields. For example, two databases A and B may contain two shared fields {*Name, Address*}, and a similarity measure (or distance metric) can be defined on both. For instance, both *Name* and *Address* could be compared in terms of string similarity: two addresses that share some words or that have similar spelling would be considered similar. Better, *Address* could be compared by using the geographical distance between the two geocoded addresses.

Combining similarity measures for different fields and establishing a decision rule results in the classification of any two records (*a,b*), where $a \in$ A and $b \in$ B, as *match* or *non-match*. If this procedure is used with a database C on itself, the result is a deduplication algorithm. In this case for any two records (c_1,c_2), $c_1 \in$ C and $c_2 \in$ C, the algorithm will establish if it is a *duplicate* and thus it will identify groups of duplicates.
When the goal is the deduplication of records, *all* the fields in the database can be used to make comparisons. However, when the goal is record linkage – and therefore two different sources of data are to be linked together – the databases may include other information in addition to some shared fields. The non-shared fields should not be discarded when they include information that is useful for matching.

Linking PATSTAT to company databases corresponds to this latter scenario. The PATSTAT database[2] includes patent information for 41 million inventors associated with 73 million patents and is a useful source of micro-data related to innovation. However, its usefulness is constrained by three problems. We list them below[3].

1. PATSTAT is noisy:
The data we find in PATSTAT is close to its raw state, and has thus not undergone a thorough process of standardization. Therefore, a number of problems surface related to data quality: incorrect spelling of names, non-standardized addresses, misplacement of address into the name field, wrong country assignment, missing data, etc. Typically, these are problems when they make it difficult to use the data, as in the case of a person without a country assigned to them. In other cases, such as when the address appears inside the person name, a thorough cleaning procedure can be helpful in getting both a clean name, and a non-missing address. However, countries have different standards in names and/or addresses, and these should be taken into account, for example when searching for legal identifiers (such as GmbH, Ltd, and the like). We can interpret every record in PATSTAT as the result of a perturbation of the real data. If we think of these perturbations – the noise – as random occurrences, then they are not too important. Noisy data can be manipulated with cleaning/standardization algorithms to make it more usable.

[2] We use the October 2011 version of PATSTAT.
[3] Improving the quality of the patent data is useful for many other kinds of research. The fact that PATSTAT is rather noisy is not so much of a problem once the appropriate cleaning procedures are implemented. See Herzog *et al.* (2007) for an overview on the data quality and record linkage.

2. PATSTAT contains ambiguous duplicates of different nature:
One of the problems related to its practical use is that one applicant may not be recognized as a unique entity for all the associated patents. In other words, multiple instances of the same entity may be recorded as separate applicants in PATSTAT. The task of being able to tell which entries in PATSTAT correspond to the same entity is referred to as data disambiguation or data deduplication. Additionally, there is no way to easily understand the nature of a PATSTAT applicant. There is no unequivocal way to identify persons and distinguish them from companies or other organizations like universities and foundations. To put it simple, they look roughly the same. Identifying the companies inside PATSTAT is important because it is a waste of time to look for a link between a person name and a company, and it may result in an overall decrease of the accuracy of the algorithm.
Data ambiguity is an important issue, and an algorithm for scalable and flexible disambiguation of PATSTAT is implemented in Huberty *et al*., 2013a.

3. PATSTAT is a poor source of company information:
Another problem associated to the use of PATSTAT for innovation research is that little information on the applicants is available in the database. Even if we could identify all the companies inside PATSTAT, we would have no information other than their name, their address (when not missing), and the classification codes of their patents. Often we can only deduce their legal form. Therefore, we need a way to overcome ambiguities and make the data richer and meaningful for research.
This paper is focused on this last problem. Our aim is to connect PATSTAT to sources of rich company data that allow using the wealth of company information available in many commercial databases. We devote a section of this paper to compare our algorithm to two other algorithm that are used to link PATSTAT to other sources of information

Examples

The three problems outlined above can be easily exemplified by looking inside PATSTAT. Example 1 and 2 report most of the useful applicant information we find in PATSTAT. The names have only been processed with a mild cleaning procedure at this step.

Example 1: Name ambiguity and noisy data.

ID	Name	Country	Address
8207275	ELECTROLUX APPARECCHI PER LA PULIZIA S P A., PESCHIERA BORROMEO, MILANO, IT	IT	
8207446	ELECTROLUX HOME PROD CORP.	BE	
8207459	ELECTROLUX HOME PRODS COPORATION N V.	BE	
8207492	ELECTROLUX HOME PRODS CORP N V.	BE	Raketstraat 40,1130 Brussel
8207500	ELECTROLUX HOME PRODS CORP N V., ZAVENTEM	BE	
8207552	ELECTROLUX HOME PRODS N V.	BG	Zaventem
8207843	ELECTROLUX ZANUSSI ELECTRODOMESTICI S P A.	IT	
8207869	ELECTROLUX ZANUSSI GRANDI IMPIANTI S P A., PORDENONE, IT	IT	
8207891	ELECTROLUX ZANUSSI VENDING S P A., BERGAMO, IT	IT	

In Example 1 we see that at least 10 different PATSTAT IDs are associated to what appears as a single entity (probably related to a large Swedish multinational company) in 3 different countries[4]. Note that the data is not of the best quality: not only are there some typos/errors in the names or in the country of origin, but also the address may appear in the name, if it is not missing altogether.

Example 2: The extent of applicant information make it difficult to categorize them.

ID	Name	Applicant	Inventor	Country	Address
10983477	GOBBI FRATTINI, PAOLO, GIUSEPPE	1	0	IT	
10983517	GOBBI GIUSEPPE	1	1	IT	
10983519	GOBBI GIUSEPPE C. S N C.	1	0	IT	Via Ancona, 5 […]
10983437	GOBBI, CRISTINA, & 20097 SAN DONATO MILANESE, MILAN, IT	0	5	IT	
10983644	GOBBI, SANTO RENZO, & 27040 ARENA PO, IT	0	3	IT	

It is in general difficult in Example 2 to distinguish person names from companies or other kinds of institutions such as universities. The available information does not allow a classification of "GOBBI FRATTINI, PAOLO, GIUSEPPE": does it refer to two/three persons, or one company, or a person with a middle name, or what else?

In general, we can probably tell that all the entries in Example 1 refer to the same entity, or that the third row in Example 2 is certainly a company, but it is not trivial for a computer to get to the same conclusion. The previously mentioned disambiguation algorithm (Huberty *et al*., 2013a) recognizes both companies and persons as unique entities based on the coauthorship patterns. It can also be used for record linkage (see Huberty *et al*., 2013b) with the caveats that apply to the standard record linkage algorithms.

In this paper, our concern is to find matches for the companies in PATSTAT. This is a dual problem, because it involves *record classification* as well as *record linkage*. Therefore, we deal with the matching challenge separately and do not use the disambiguation output of earlier work (Huberty *et al*., 2013a). We also want to make use of the wealth of information inside the company database to avoid making meaningless links between the two data sources.

We introduce REMERGE, a flexible algorithm that allows to
1. identify the person names in PATSTAT
2. link the non-person names in PATSTAT to a company database[5] such as Amadeus/Orbis

[4] Note that we may be looking at an example of company acquisition: that is an additional problem that we do not analyse in our work.

[5] Since multiple entries in PATSTAT can be matched to the same company, this algorithm also disambiguates the companies inside PATSTAT.

Data and terminology

We use the October 2011 version of the PATSTAT database, considering patent applications for EU-27 filed since 1990.

The source of company data is less relevant. Any company database that includes information such as name, country of origin, address, sector, number of subsidiaries, number of employees, website address, and value of intangibles can be matched to PATSTAT. Examples of such databases are Compustat, D&B Database, Hoovers, Orbis. We use the Amadeus/Orbis database as available through Wharton Research Data Services[6].

In our further discussion, we will make use of some terms in a standardized fashion to ease the later discussion.

- *PATSTAT entity*: any person, company, or other type of organization that appears in PATSTAT as an applicant, inventor, or both. There is no clear-cut differentiation between persons, companies, and other organizations in PATSTAT, so we will adopt this general term to refer to all of them. A PATSTAT entity can obviously be associated to many patents (PATSTAT as an applicant, inventor, or both), but in our definition it is associated to a unique name (see below)

- Every PATSTAT entity has *at least* one *PATSTAT ID*, which is a code that identifies that entity inside the database. Every PATSTAT entity can be an inventor, an applicant, or both, but companies are never inventors[7]. We thus only consider PATSTAT entities that are not inventors. A PATSTAT entity can have multiple PATSTAT IDs[8].

- *PATSTAT name*: the name associated to the PATSTAT entity. We group together all the PATSTAT records that share the same name and consider them as a unique entity. The clean-up procedure applied to the PATSTAT names has thus the immediate effect of reducing the variation in the PATSTAT names, and therefore the total number of PATSTAT entities for which a match is searched.[9]

- *Company*: a record in the company database. Matching PATSTAT to company databases is thus the task of finding a company that matches a PATSTAT entity.

- *Pairing*: any PATSTAT entity-company comparison.

- *Candidates*: for a PATSTAT entity, it is the subset of all pairings that are most likely to include a true match. We typically use a string distance measure to filter the candidates.

---

[6] Data from WRDS was last downloaded on March 6th, 2014. The list of extracted variables is in Appendix 1.
[7] A PATSTAT ID of a company may be associated to an inventor number greater than zero because of a typo, but that would only happen for a very limited number of patents without affecting the overall count.
[8] In PATSTAT, this is the *person_id*. We avoid using the term *person id* to make it clear that we do not have information on whether a company is an individual or a company.
[9] Recognizing which records in PATSTAT are the same entity is the task of deduplication. In this paper, we apply a simple, deterministic deduplication rule (i.e. we group together all the records with the same name), but machine learning can be used for this task as well, see Huberty *et al*. (2013a).

- *True/False match*: the truth value of any pairing.

- *Training set*: a sample of PATSTAT entities and related candidates for which we identify the truth value manually. The training set is used to find the best performing model, which then will be used to predict the truth value of all other pairings[10].

- The regression will use some *explanatory variables*: these are the features that we get directly from the data or that we obtain by elaborating the data. Predictions on the pairings will be based on a decision rule created from these variables.

- *Record linkage or matching*: the task of determining whether two records from two different sources are really the same, i.e. they are a *match*.

Therefore, in other words: for every PATSTAT entity, we look for possible true matches by selecting some candidates out of all possible pairings. We select the candidate that has the highest estimated probability of being a true match. We estimate this probability for all candidates by applying the model that works best in the training set.

[10] For more details, see Appendix 2.

Record linkage: overview

Record linkage algorithms usually require the two sources being merged to be compared based on the fields they share. The theorization of record linkage is by Fellegi and Sunter (1969). If two databases containing persons' information are to be merged, we would compare names, addresses, and whatever other information is available so that a summarizing similarity measure between two records can be defined. However, sometimes this approach is too restrictive, as it may not use the full wealth of available information. If we only compared records based on the shared fields between PATSTAT and company databases, we would end up using only *Name* and *Address* (the latter being missing in PATSTAT for a large share of the records). However, this is not sufficient to obtain relatively high accuracy of the matching, for two main reasons:

(1) Many applicants in PATSTAT are not companies, or are companies that do not exist anymore. However, PATSTAT does not include information on the nature of the applicant, meaning that we cannot filter the database *a priori*.

(2) The large majority of companies in Amadeus (or other databases) does not apply for patents. However, Amadeus does not include information on patents (Orbis does include information, but that is the result of a relatively intransparent matching procedure to PATSTAT, thus the information is not native to the database).

Trying to link PATSTAT to Amadeus based on the shared fields alone would result in many incorrect person-company links. In order to take into account that not all applicants are companies, and that not all companies are applicants, we could construct a concordance measure between patents' IPC codes and company sector codes – after all, it is unlikely for a restaurant to apply for a patent related to solar panels.[11] We could also think of other similarity measures between the variables in PATSTAT and Amadeus. But that would add steps to the overall procedure, and decrease its flexibility.

REMERGE does not require records of the two databases to be compared on the basis of the same variables. This means that we can use information in addition to just the name and geography of records without having to resort to the construction of ad-hoc similarity measures between fields of different nature. Our approach to record linkage is a supervised learning algorithm that uses a penalised regression to estimate the probability that any PATSTAT entity matches a company. The regression will use all the available variables, their interactions, and will result in a transparent decision rule that optimizes the predictive performance. REMERGE is most useful when a link should be established between databases that do not overlap either vertically (i.e. some records are not supposed to have a corresponding link) or horizontally (i.e. records in different databases carry very different information).

[11] See Lybbert and Zolas (2012) for a detailed discussion on IPC concordances.

Record linkage: the algorithm

The algorithm proceeds in a few steps. There are six main steps, each of them divided in other smaller steps.

1. Clean and geocode the data
2. Aggregate all the PATSTAT IDs that share exactly the same name
3. For each PATSTAT entity, find the candidates within a certain block of data, based on the string distance between the names
4. For each candidate pairing, obtain all selected variables from the two databases, and calculate new ones (e.g., interaction variables)
5. Run the Lasso penalized regression on a hand-curated training set
6. Use the estimated coefficients to classify all other data

1. Clean and geocode the data

We start by cleaning the data. The same procedure should be applied to both sources of data. The clean-up process includes diacritic removal, case standardization, and abbreviation standardization. Legal identifiers written in their extended form are substituted with their acronym. When addresses are in the name field, the algorithm attempts to identify them and separate them from the name. Finally, we geocode the addresses at the city level.
The technical details as well as the potential customisations are explained in the code documentation.

2. Aggregate all the PATSTAT IDs that share the same name

In a first run, PATSTAT IDs that share exactly the same name are aggregated. This has the main effect of greatly reducing the number of PATSTAT entities for which a match needs to be found.
We only perform this name-based aggregation on PATSTAT, but not on the company database. There are two reasons for this.
First, it is very difficult to tell apart two PATSTAT entities if they share the same name (and country). We assume they are the same entity. This strategy could incorrectly merge two companies that have been patenting in different fields. This is a source for additional noise in the matching process[12], but we have too little information to proficiently improve this part of the algorithm.
The second reason why we only do name-aggregation on PATSTAT is that commercial company databases usually go through a process of standardization and should not contain duplicates. If two companies have the same name – this is a rare occurrence compared to how often it happens in PATSTAT – a look at their sector or their address will almost always make it apparent that they are not actually the same company. Further deduplication of records in

[12] This is why the deduplication of PATSTAT companies would have made matters worse, if applied before the record linkage.

company databases implies having a definition of what a company is, but this question may not have an easy answer.[13]

3. Find the candidates

At this stage, we have cleaned and geocoded both sources of data, and aggregated PATSTAT so that it is only composed of unique names. We now need to perform a filtering of the data to avoid making too many comparisons. It is impractical to compare all PATSTAT names to all companies. Therefore we need a blocking strategy that allows to only make the most relevant comparisons and identify the true match for a PATSTAT entity from a reduced set of companies.

For every PATSTAT name, we filter out all companies that do not correspond to the following criteria:

the country of origin matches that of the PATSTAT entity

AND

after removal of the spaces from both names, the first two characters coincide	OR	the first three characters of one of the first three words in the PATSTAT name coincide with the first three characters of one of the first two words in the company name[14]

We then take this set of companies and we compute the Levenshtein ratio and the Jaro-Winkler string distance measures for all the pairings of the PATSTAT name with the company names. For both name distance measures we sort the companies from the most similar to the least similar, and we keep the ten company names that are most similar to the PATSTAT name. Finally, we take the union of the two sets. This means that for every PATSTAT entity, we now have less than twenty companies among which a match can be found. As we previously mentioned, we label these companies the *candidates* for the PATSTAT entity.

This filtering process has a great advantage in that it allows to reduce the number of comparisons, speeding up the process of record linkage. However, it can also be source of a lower recall.

In Example 3 we see the result of this process for a French PATSTAT entity: *eurocopter france inc*. There are fourteen candidate companies for *eurocopter france inc* based on their

[13] Consider parent-child relations among companies, but also joint-ventures, mergers and acquisitions, spin-offs, and so on. These events happen over time, and they complicate the task of recognizing unique entities in the company database. This is a reason why we do not deduplicate the company database.

[14] In the UK, the large number of companies makes it unfeasible to consider the first two words in the company name, so we only consider the first word. The blocking strategy may result in no candidate being found, in which case we discard the PATSTAT entity.

name distance, which we measure in two ways: the Levenshtein ratio (*lev_ratio*) and the Jaro-Winkler distance[15] (*jw_dist*). We sort this list by *jw_dist*.
Now, by looking at the table we see that *eurocopter* is the company that matches the PATSTAT entity. However, if we reasoned in terms of string distance alone, then we would choose *europrotect france sa* because it has the most similar name to *eurocopter france inc*. Instead, the true match *eurocopter* has a Levenshtein ratio of 0.35 and a Jaro-Winkler distance of 0.10 and is only the seventh closest candidate.

In other words, while we were able to drastically reduce the number of comparisons from millions to just 14 in this case, there is still a lot to do in order for the true match to appear as the best candidate for *eurocopter france inc*.

Example 3: Example of candidates list for a single PATSTAT entity. Smaller *numbers for* lev_ratio *and* jw_dist *correspond to more similar names. The true match is in bold – note that it is* not *the company with the most similar name to the PATSTAT entity:*

PATSTAT name	PATSTAT legal	COMPANY name	COMPANY legal	lev_ratio	jw_dist
eurocopter france inc		europrotect france	sa	0.28	0.07
eurocopter france inc		euro performance		0.30	0.08
eurocopter france inc		eurotherm france		0.24	0.09
eurocopter france inc		euro crm france		0.28	0.09
eurocopter france inc		euroconcept in	gie	0.31	0.10
eurocopter france inc		eurocopter training services		0.31	0.10
eurocopter france inc		**eurocopter**		**0.35**	**0.10**
eurocopter france inc		europatech france		0.26	0.11
eurocopter france inc		eurocen france		0.26	0.11
eurocopter france inc		europartner france		0.23	0.13
eurocopter france inc		eucopower france		0.24	0.15
eurocopter france inc		eurocap france		0.26	0.15
eurocopter france inc		eurocir france		0.26	0.15
eurocopter france inc		cooper france finance	snc	0.24	0.24

4. Retrieve/calculate the variables for matching

At this point, we have a single table for every country that contains the PATSTAT names and the list of candidate companies for every PATSTAT name (Example 3 shows a portion of this table). We now add to the table all the information we have in PATSTAT and in the company database to obtain a more detailed view on the pairings. Every row in a table is a pairing between a PATSTAT entity and a company, and we have information such as:

- PATSTAT entity information:
    - how many times the PATSTAT name appears in a patent as applicant,

[15] By string distance we mean a number between 0 (the two strings are the same) to 1 (completely different strings). There is a variety of string distance measures. We choose the simplest one (Levenshtein ratio) and a somewhat more elaborated one (Jaro-Winkler).

    - the list of IPC 4-digit codes of the related patents,
    - the last year in which that PATSTAT name appeared,
    - whether the name was abbreviated during the clean-up phase,
    - the geographical coordinates if available.
- Company data such as sector and number of employees.

We then manipulate this data to obtain additional information that could be useful in determining which one of the candidates is the true match, and if a true match can be found. The explanatory variables are then going to be used later in the regression.

What follows is a list of variables that we create:

- *perfect_match* is an indicator that is 1 if the PATSTAT name and the company name are exactly the same
- *legal_jw* is a string distance measure on the legal identifiers extracted from the names (if any)
- *name_less_common_jw* is a string distance measure on the names after the removal of words that are common inside PATSTAT and company names. The list of common words is calculated in-sample, based on the PATSTAT and company names
- *metaphone_jw* is the distance between the two names' sound when spoken out loud
- *ps_web_jw* is the string distance between the PATSTAT name and the company website (if any)
- *min_jw_of_alt* is the maximum string distance of the PATSTAT name with the *other* company names
- *avg_freq_am* and *avg_freq_ps* are the average frequency of words in the Patstat name and the company name. We may want to discount a strong string distance if the company name is made up of very frequent words
- *geo_dist* and *geo_cat* are variables measuring the geographical distance between addresses. geo_cat is a categorical variable obtained from geo_dist
- *bracke*t is a categorical variable in 4 levels that assigns a company to a revenue bracket. Higher values correspond to larger revenues.
- *sector_sim_max* is the maximum distance[16] between a company's sector and a group of IPC codes. This is useful because it tells us how frequently a company in some sector holds patents with those IPC codes
- interaction variables: we interact sectors and countries, countries and estimated revenues, countries and geographical distance to grasp possible peculiarities for specific countries

Finally, early testing of the algorithm revealed one of the main problems in the record linkage of PATSTAT with company databases, that is the impossibility of determining with absolute

[16] It was calculated using the *unique perfect matches* between PATSTAT entities and companies, i.e. the list of PATSTAT entity names that perfectly correspond to exactly one company name. With this list, we have the sector and IPC codes and we build a contingency table with the relative frequencies

certainty whether a PATSTAT entity is a company or not. Hard-filtering the data according to some criterion (e.g. removal of entities with no legal identifier) may result in a very precise subset of the original data because all remaining entities will be companies. However, it also leaves out a lot of companies that did not pass the test.
We preferred to apply a very mild hard-filter[17], but we additionally also estimate the probability that a PATSTAT entity is a person, and we use this to calculate *is_matchable*. This variable is the estimated probability that the PATSTAT name is not a person. We use PSCLASSIFY to perform this filtering and estimation task.

[17] We remove from the data all the PATSTAT entities that appear as inventors but not as applicants in a patent. Since one entity can appear in multiple patents, this hard-filter does not remove a name if there exists a patent in which that name appears as applicant. By contrast, it removes a name that always appears as inventor-not-applicant.

PSCLASSIFY is a small classification algorithm that allows to estimate the probability that a PATSTAT entity can be matched to a company. In fact, it is impossible to establish with certainty if a PATSTAT name belongs to a person, a company, or some other kind of institution.

PSCLASSIFY estimates Pr(name is a person) and sets this to 1 when the name includes a country-specific legal identifier at the beginning or at the end of the name. Then, it uses the following variables, and their interactions, in a L1-penalised logistic regression to compute the estimation for all other names:

- country of origin
- applicant sequence number (companies are usually in the first places of the applicants lists)
- word count of the name
- average word length of the name
- was the name abbreviated during cleanup?
- is the name only made of letters or are there numbers or special characters?
- is there a legal identifier inside the name?
- is a legal identifier of a foreign country inside the entity's name?
- is a common name in the name?

The following table shows the result of this estimation procedure for some PATSTAT names. *is_matchable* will be later used as a variable inside REMERGE.

patstat_name	**is_matchable**
fusco maria antonietta	0.03
andersen irma	0.09
fritsch hans joachim	0.29
gartnereibedarf asperg eg	0.37
franz plasser bahnbaumaschent ind	0.65
hans josef dahmen steppomat textilmaschenbau inhaber peter ringhut	0.81
frandsenlyskilde as braedstrup dk	0.81
gascoigne melotte b v emmeloord nl	0.98
water savers bv	1
franz plasser bahnbaumascher ind gmbh	1
frontera azul systems sl	1

We also calculate the interactions between countries, sectors, IPC codes, and other variables, and we end up with a total of around 9000 variables.

5. Run the regression on the training data and select the best model

We build the training set by randomly extracting 1013 PATSTAT entities, each of them having a list of candidates. We manually look for true matches in the data – this procedure is facilitated by a small script that loops over the sample of PATSTAT entities and displays the candidates, making the identification of the possible true match straightforward. Since we assume that there exists at most one true match for any PATSTAT entity, the resulting training set will mostly be made of zeroes.

We use the training set to fit the L1-penalized logistic regression. This kind of model involves automatic variable selection and shrinkage (like the Lasso of Tibshirani, 1996). As a result of this procedure, most of the variables get discarded and just a few – the most useful for prediction – remain in the final specification of the model. The Lasso model uses a penalty parameter that manages the strength of the variable selection mechanism. The model selection procedure involves a fundamental parameter, i.e. the Lasso penalty parameter that controls the strength with which variables are shrunk and selected: higher values correspond to fewer variables in the final model.

However, similarly to a classification problem, a probability cutoff must also be chosen according to an objective function to be optimized. Every pairing with an estimated probability of match lower than this cutoff should be discarded as a non-match, and every pairing with an estimated probability higher than the cutoff such that no other pairing has a higher estimated probability should be considered a match. For every PATSTAT entity, we only consider the candidate that is associated to the highest estimated probability of being a match.

In a typical classification problem, we would take the probability cutoff out of the picture by considering the area under the ROC curve as the objective function. However, we choose to consider the harmonic mean between precision and recall (F1-score) as our objective function as it does not depend on the true negatives we managed to identify. This function does however depend on the probability cutoff. Therefore, we select the Lasso penalty parameter that corresponds to a model that allows for the highest out-of-sample F1-score to be reached for some probability cutoff.[18]

Since this is a prediction exercise, we split the training set in two. One half is used to estimate 100 models at varying levels of the Lasso penalty parameter. The other half is used to test the out-of-sample prediction performance according to the above mentioned function.

[18] There are at least two arbitrary choices made here. The first is the choice of the F1-score using equal weights for precision and recall: it may well be the case that precision is more important than recall, in which case a modified F1-score, or a different function altogether should be used. Also arbitrary is the choice of considering the maximal F1-score. We could have picked the model that optimizes the *average* F1-score across probability cutoffs, perhaps under the belief that this results in more reliability.

The resulting best model selects around 300 variables for the estimation of the probability that a pairing is a true match.

6. Use the estimated coefficients to classify all other data

With the complete list of relevant variables selected by the regression we estimate the probability of being a match for all pairings in the data. Finally, we discard all company-candidates for a certain PATSTAT entity for which there is a company-candidate with a higher estimated probability of match.

Other approaches to PATSTAT record linkage

Introductory remarks

In general, linking PATSTAT to company databases involves a great deal of effort to be devoted to the clean-up of the data. This effort is justified by the fact that the procedures that have been used up until now have focused on the exact matches, or on the similarity of the names only. Therefore, having a very good clean-up procedure allows for the identification of more matches.[19] This applies to the OECD (Thoma et al., 2010) and EPO/OHIM (2013) procedures explained below, but also to other efforts such as Lotti and Marin (2013). In the following section we will see how REMERGE does benefit from good data-cleaning procedures, but is less dependent on them than the other algorithms.

Comparison with the EPO/OHIM procedure

We discuss similarities and differences of our algorithm with the one used by EPO/OHIM in their report *"Intellectual property rights intensive industries: contribution to economic performance and employment in the European Union"* (September 2013[20]).

The EPO/OHIM group recognize the very same problems that we outlined above (ambiguity, lack of information on the applicants) and develop an algorithm to link PATSTAT to Orbis. We summarize it here, noting the similarities and differences with our algorithm.

Clean-up of the names in both sources.

This stage is very similar to our algorithm, especially in terms of the strategy used to deal with legal identifiers. However, a difference is that our algorithm runs only once and is the same for all countries. Another difference is that EPO/OHIM compiled a list of frequent or non-distinctive words and *removed* them from the name field. First, this procedure was labour intensive and non-automatic. Second, some information may be lost in the process. Instead, we adopt an automated procedure that removes the common words and outputs a cleaned version of the names, on which we calculate the string distance and obtain *name_less_common_jw* as an additional explanatory variable. Thus, we are able to remove common words without affecting the information content of the original names.
EPO/OHIM recognize the difficulty in establishing whether a PATSTAT name corresponds to a natural person. They deal with this problem by using the comma as delimited and splitting the PATSTAT names in two. While it is true that PATSTAT names for natural persons are of the form *"Last Name, First Name"*, this does not always apply. Instead, we run PSCLASSIFY on the names to give a probabilistic assessment on the PATSTAT names.

Linkage with Orbis.

EPO/OHIM go through a number of steps to find the matches with Orbis companies. The first phases look for matches by using the names only. For all the PATSTAT names that were matched to a multiplicity of Orbis names, further checks such as a ZIP code comparison are run so that the best-fitting company is ultimately assigned to the PATSTAT name.

[19] Multiple matches are usually dealt with manually.
[20] http://ec.europa.eu/internal_market/intellectual-property/docs/joint-report-epo-ohim-final-version_en.pdf

All of the steps are mechanical, and thus result in a match/no-match deterministic decision. Finally, an additional manual matching phase was carried out to reduce the bias induced by the fact that large companies tend to be easier to link. This certainly also holds for REMERGE, but the bias is somewhat reduced because REMERGE does not solely base its probabilistic assessment on the perfect name concordance like the EPO/OHIM algorithm does. In other words, REMERGE can link together two names even if they are different, if other variables point to them as being originated by the same entity.

Summary

While the EPO/OHIM algorithm surely results in very precise matches, it is also very dependent on the structure of the Orbis database and requires a lot of manual work. Our algorithm, instead, is able to give an assessment of the probability that any pairing is a match and works with any kind of company database.

Finally, we point out that EPO/OHIM only considered patent applications filed between 2004 and 2008, whereas we extract data from 1990 to 2011. There is one main consequence from this fact: their coverage numbers are going to be much higher than ours, because it is much more difficult, on average, to find a match for a company that applied for a patent in the ‘90s and may not even be in the company database.

Comparison with the OECD procedure

OECD (Thoma *et al*., 2010) adopts an algorithm that uses the Jaccard string distance measure on the cleaned names to identify the approximate matching between PATSTAT names and Orbis companies. Again, the cleaning phase is similar to both our algorithm and the EPO/OHIM algorithm. The approximate matching, instead, relies uniquely on the string distance between names and does not use other information. In this sense, it shares a similarity with REMERGE in that it performs fuzzy matching, but also with the EPO/OHIM methodology in that it only uses string distance. However, as we will see in the next section, REMERGE outperforms algorithms that only rely on string distance, and additionally offers flexibility for further improvements.

Performance

Comparison with simpler linkage algorithms

We compare REMERGE to a simple algorithm solely based on the simplest string distance measure (the Levenshtein ratio of the names[21]). This algorithm is labelled *lev_clean* and it uses the same cleaning procedure used by REMERGE. Once the names have been cleaned, we apply the same blocking strategy as in REMERGE. Afterwards, for every PATSTAT entity we select the company with the most similar name. If there are multiple companies with the same name, we pick the match at random. Finally, we choose a threshold for the string distance measure under which the match is discarded (REMERGE uses the estimated probability to do the same thing). For instance, in Example 3 *lev_clean* would pick *europrotect france sa* as a match for *eurocopter france inc* because it has the most similar name according to the Jaro-Winkler string distance measure.
Note that *lev_clean* is somewhat similar to the OECD procedure outlined in the previous section, but it should not be considered as a faithful reproduction of that procedure.

We compare the three algorithms based on their performance on a random sample of 200 PATSTAT IDs[22].
We define True Positive, False Positive, True Negative, False Negative as follows:

- True Positive (TP): the correct company has been assigned to the PATSTAT entity
- False Positive (FP): the incorrect company has been assigned to the PATSTAT entity (the correct one can be another company or no company at all)
- True Negative (TN): the algorithm correctly avoids matching with the PATSTAT entity
- False Negative (FN): a company should have been matched to the PATSTAT entity, but no company was matched instead.

We use these to compute precision, recall, and F1 score at different thresholds.

- Precision $= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$
- Recall $= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$
- F1 score $= \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$

[21] The results hold in general even when comparing algorithms that use more complex string distance metrics.
[22] Recall that we consider as PATSTAT entity the set of PATSTAT IDs that share the same name. Since the different algorithms clean the name differently, the PATSTAT entities will also be different. A random set of PATSTAT IDs is thus the only neutral choice.

Figure 1: Precision, Recall, and F1 score of REMERGE *and fuzzy matching. Exact matching corresponds to fuzzy matching with cutoff set at 1.*

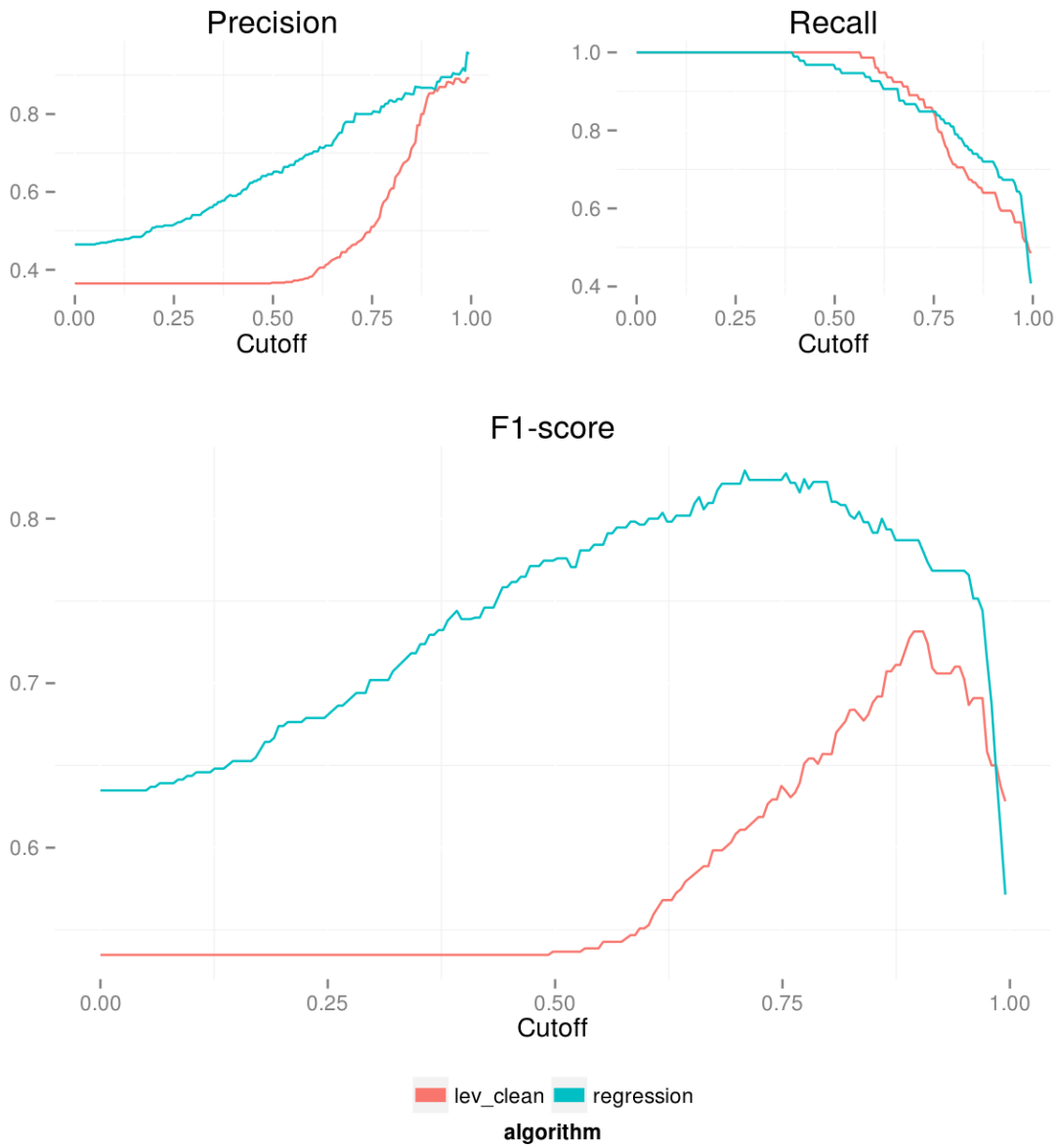


Figure 1 indicates that REMERGE is vastly superior to *lev_clean*. Also note how for a wide range of threshold values (from around 0.5 to around 0.9) REMERGE shows an F1 score of more than 0.85. Since the F1-score is the harmonic mean of precision and recall, we deduce that the algorithm has a rather good performance in absolute terms.

The fact that REMERGE does better than *lev_clean* does not come as a surprise: REMERGE is a generalization of *lev_clean* because it includes the Levenshtein ratio into the regression. The general point here is that a regression-based record linkage such as REMERGE will do better than *any* algorithm that is limited to only 1 comparison variable, if both are applied to the same data.

Coverage

We have seen that REMERGE has a good performance in terms of precision and recall. This means that on one hand, what we obtain as an estimated match is very likely the true match (high precision), and on the other hand, we should be able to extract a large proportion of the real matches, missing only a few (high recall). We now set a threshold value of 0.7 for the estimated probability of matching (according to Figure 1 this threshold results in more than 85% precision and more than 80% recall), and see how many PATSTAT IDs and how many companies are affected by the matching algorithm.

Figure 2 shows that around 40% of the patents are linked by REMERGE to a company. This figure varies across countries, and goes beyond the 50% mark for Denmark.
In Figure 3, we can see the percentage of companies for which REMERGE found a PATSTAT entry. The numbers are much lower and not uniform across countries. This can be largely explained by the fact that company databases include all kinds of companies, and not only those that are patenting their innovations. We also note that countries have different economic/business structures that might explain the differences we are observing.

Figure 2: Coverage of the linkage on PATSTAT

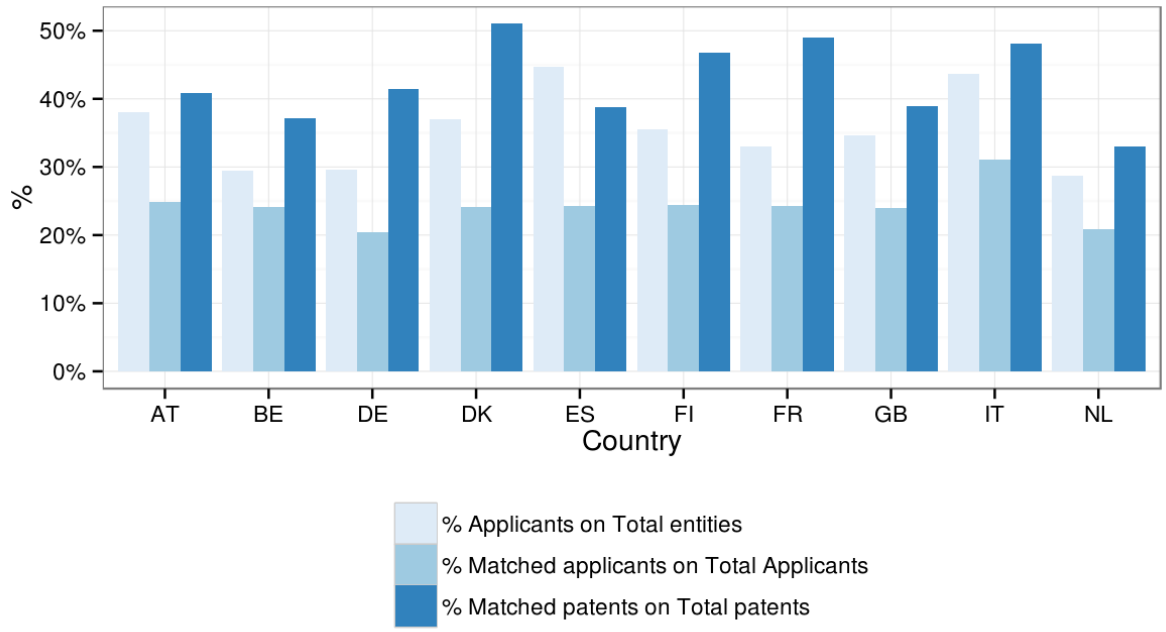


Figure 3: Share of companies for which REMERGE found a matching PATSTAT ID

We should consider Figure 2 and 3 together: a relatively small number of companies is responsible for the majority of the patents. For example, consider the case of Belgium: less than 30% of the PATSTAT IDs are applicants, and around 25% of the applicants have been matched to a company. In turn, we assigned patents to a very small share of all the companies in Belgium. We deduce that more than 40% of the patents in Belgium are to be attributed to a very limited set of companies.

We finally highlight the differences across sectors in Figure 4. We see that some sectors have a larger share of companies matched to some PATSTAT ID, as is to be expected. For example, five percent of manufacturing companies have been linked to at least one patent application.

In Figure 5 we display an example of data visualization that can be done after the record linkage. A company database like Amadeus includes relatively precise address information, unlike PATSTAT. Therefore we show green and fossil patents' geolocation on a map (based on patent categorization in Lanzi *et al*., 2011).

Figure 4: Percentage of companies matched to a PATSTAT ID, by sector

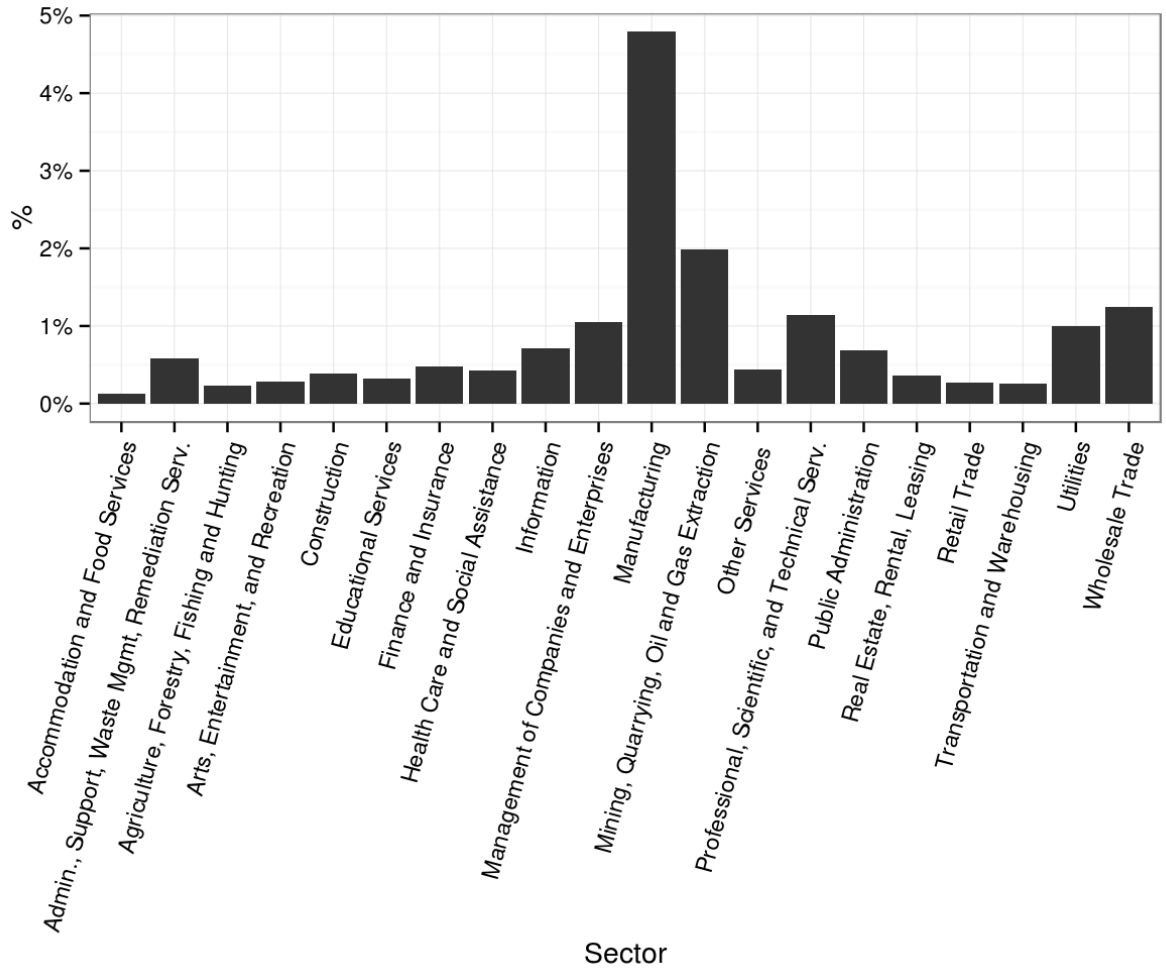


Figure 5: Green and fossil patents matched to a company, by location, 1990-2011. Black circles refer to fossil patents. Larger circles are associated to locations with higher patent counts. Locations are based on patenting companies' addresses.

EEE-PPAT classification comparison

Linking companies to PATSTAT patent applicants effectively involves the identification of companies. One problem with the figures reported above is that we only could see which links REMERGE found, without being able to tell what the numbers should have looked like in an ideal situation. We can partially overcome these problems by using the EEE-PPAT table.

The EEE-PPAT table (Du Plessis *et al*., 2009) provides information on the applicants. It categorizes patentees into private business enterprises, universities / higher education

institutions, governmental agencies, and individuals. It is developed by ECOOM in partnership with Sogeti and is external to PATSTAT. Reported quality levels of 99% are obtained in terms of completeness and accuracy. If we take the EEE-PPAT table as the ground truth, we can evaluate REMERGE in terms of how it classifies PATSTAT entities into companies and non-companies. Satisfactory results of REMERGE when compared to EEE-PPAT are a necessary (but not sufficient) condition for good record linkage performance. The EEE-PPAT table can additionally be used as:

- a hard filter before the algorithm is run, to only keep the applicants that are labelled as companies;
- a variable inside the regression very much like PSCLASSIFY;
- a filter after the regression, to discard the links between persons and companies.

We decided to leave the EEE-PPAT table out of the algorithm. This way, REMERGE does not depend on external information and can be applied with no delay as soon as a new version of PATSTAT is released.

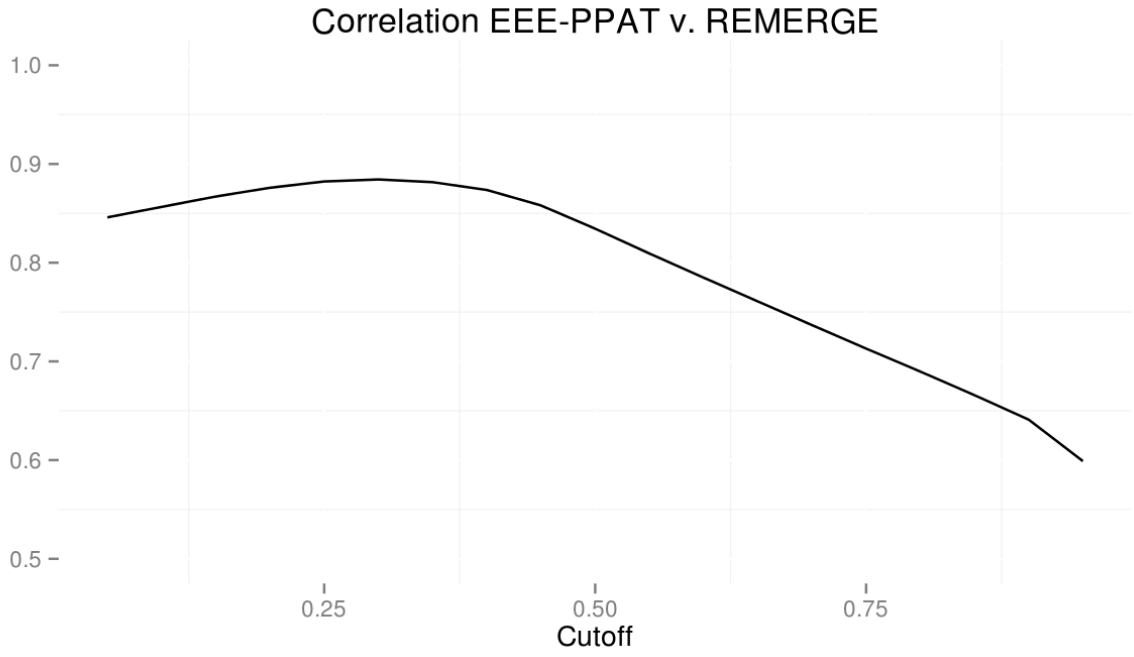
We now compare the company assignments made by REMERGE to the EEE-PPAT table. In Table 4, we report the categorization made by EEE-PPAT table on EU27 data from 1990 to 2011. Since REMERGE only distinguishes between companies and non-companies, we also aggregate the data and show the total counts. There will be mismatches between EEE-PPAT and REMERGE. In particular, REMERGE will classify as companies some entities that EEE-PPAT classifies as something else, and vice-versa.

Table 4: EEE-PPAT sector assignments for EU27

EEE PPAT sector	**Counts**
COMPANY	712121
COMPANY GOV NON-PROFIT	4430
COMPANY GOV UNIVERSITY	30
COMPANY HOSPITAL	245
COMPANY UNIVERSITY	254
GOV NON-PROFIT	19722
GOV NON-PROFIT UNIVERSITY	97
HOSPITAL	636
INDIVIDUAL	1045161
UNIVERSITY	15089
UNKNOWN	96311
N/A	3976423
	Totals
COMPANY	717080
OTHER	5153439

As previously mentioned, REMERGE may link a company to a PATSTAT entity through the attachment of an estimation of the probability of the two being a match. This probability will have the usual [0, 1] range, and this means that a cutoff must be chosen in order to evaluate the results through the EEE-PPAT table. In fact, if we choose a very low cutoff, many links will be established, but a share of them may be false matches. Instead, choosing a high cutoff results in fewer, more precise links. The choice of a decision rule depends on the objective function – e.g. higher cutoff if precision is more important. In this section, we want to look at how remerge is able to distinguish companies from non-companies, and we take the EEE-PPAT table as the ground truth. Therefore we choose the cutoff that maximizes the correlation between the REMERGE category and the EEE-PPAT sector. Figure 6 shows how the correlation changes with the cutoff in REMERGE.

Figure 6: Correlation between EEE-PPAT and REMERGE classification. The maximum of 0.884 is reached at a cutoff = 0.30



We consider the remerge output for a cutoff of 0.3 and report some results in Table 5. At this cutoff, 94.12% of EEE-PPAT companies were also classified as companies by REMERGE. REMERGE also classifies as companies almost 52% of the entities that were classified as unknown by EEE-PPAT. At the same time, 0.82% of what REMERGE classifies as non-companies were actually found to be companies by EEE-PPAT. This corresponds to 42196 entities in PATSTAT.

These numbers show that there is a large correspondence between the classification carried out by remerge and the one in the EEE-PPAT table. Obviously, this does not guarantee that all companies linked to PATSTAT by remerge are true matches, but it shows that REMERGE is *looking in the right place*: it would have been worrying if we had found that REMERGE considered as companies some entities that actually were not. On the other hand, we do not

expect to see a perfect correspondence, as Amadeus may still not include companies that only existed in the early years that we took into consideration for the linkage (1990-2011).

Finally, the EEE-PPAT table is manually curated and shows very high classification accuracy. For this reason, while REMERGE will work regardless of the availability of the EEE-PPAT table corresponding to the PATSTAT version in use, the implementation of an a-posteriori filter after matching that considers the EEE-PPAT table results is recommended.

Table 5: differences in classification by EEE-PPAT and REMERGE.

	cutoff	0.3
	company class correlation	0.884
	REMERGE	
EEE-PPAT company	company	94.12%
	other	5.88%
EEE-PPAT unknown	company	51.94%
	other	48.06%
	EEE-PPAT	
REMERGE company	company	85.88%
	other	14.12%
REMERGE others	company	0.82%
	individual	19.97%
	unknown	0.91%

Not only PATSTAT: TED – Thompson Reuters linkage

REMERGE is a general purpose algorithm, and it can be applied to other record linkage problems. It is most useful when there is information on the data that helps establishing whether there should be a link in another database. We now look at an additional record linkage problem to compare REMERGE to the simpler fuzzy/exact matching algorithms.

We received from the European Commission the database containing all the notices published on the TED[23] website between 2008 and 2012. TED is the online version of the *Supplement to the Official Journal of the European Union* dedicated to European public procurement. The TED database suffers from similar issues to those we previously identified for PATSTAT. The Thompson Reuters (TR) database on company financial covers 99% of the world's market capitalization, but that limits its scope to only 51,900 companies.

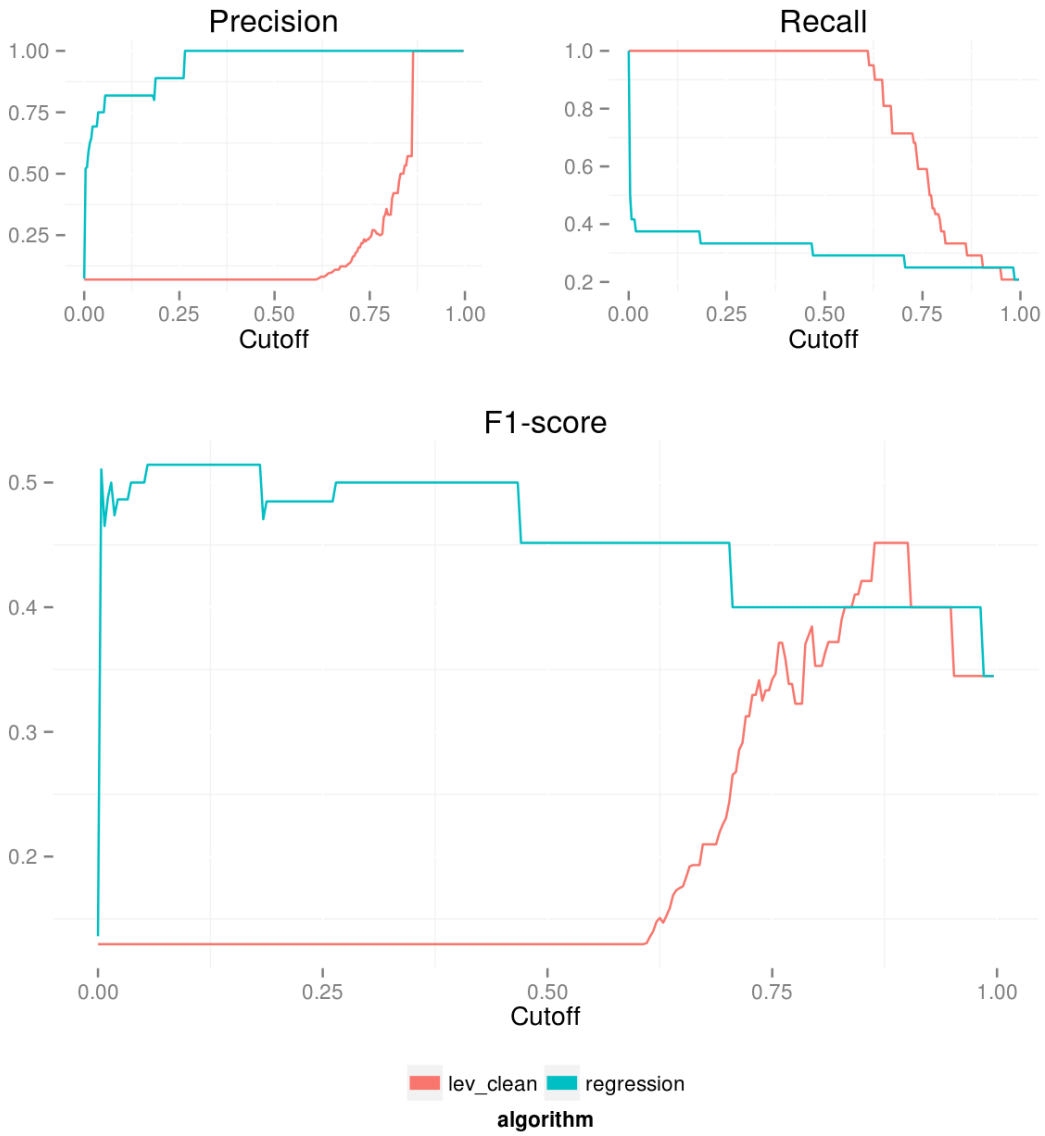
We apply REMERGE on the new data in a way that closely resembles what we did for PATSTAT. We want to compare its results with the other alternative typically used for matching, i.e. fuzzy matching or exact matching. Figure 7 reports the results of the different linkage algorithms. Again, exact matching corresponds to the fuzzy matching with cutoff at 1.

As in the earlier discussion, we note that REMERGE does better than either fuzzy matching or exact matching, but we also note that the performance is worse than with PATSTAT. This is because the task of linking TED to TR is a slightly different operation. First, since the coverage of TR is so limited, we do not expect to find a lot of matches between the two databases: for this reason, we may want to value precision more than recall. Second, since the majority of entities in TED will not find a match in TR, there may be issues with the training set (may be too unbalanced). Third, the cleaning procedure applied to TED was the same applied to PATSTAT, and this may be the reason why the performance is worse than with PATSTAT.[24]

[23] Tenders Electronic Daily, website: ted.europa.eu

[24] The TED database often includes 3 or more companies in the name field: we did not spend time on trying to separate the names as there is no fixed rule to separate them. That may be a project on its own.

Figure 7: Performance of remerge compared to fuzzy matching when applied to the TED-TR linkage.



Overall, we are still able to see an improvement of REMERGE over fuzzy or exact matching, even with no customization to the cleaning procedure.

Shortcomings and further work

There are a few critical points in the algorithm, and they reduce the accuracy of the matching between PATSTAT entities and companies. These constitute the possible avenues for further improvement.

Issues with the blocking strategy

The blocking strategy may be in some case too restrictive and hide some companies from the algorithm, inflating the False Negatives. For example, a company may be associated to multiple PATSTAT IDs, and some of them may be recorded in a different country. Or again, the blocking strategy may filter out some acronyms because they do not correspond to the full name that appears in PATSTAT, or vice versa. This problem may be solved by using a dictionary that substitute the name with the acronym, but this is not a flexible solution.
In general, problems with the blocking strategy are eased by relaxing the blocking strategy itself, but there is a trade-off with computation time that one should always bear in mind.

Issues with the regression model

The regression model uses data (the training set) derived from the set of candidates. This means that for every PATSTAT entity in the training set we have up to twenty candidate companies. When we label one of them as a match, we automatically label the others as non-matches. While this reduces the time spent on hand-labeling, it also produces a non-iid sample. This means that the estimated model is affected by the number of candidates we insert in the training data. For example, it does make a difference if we increase the number of candidates from up to twenty to up to forty even if no match was ever among the companies we added last.
Perhaps this is more easily explained by mentioning that at its current state, the model values True Negatives as much as True Positives, whereas it should not care about the True Negatives. In fact, for one PATSTAT entity we assume there can only be one True Positive. However, there are always millions of True Negatives. In any case, we do run a part of model selection using our objective function that uses our definitions of TP, TN, FP, FN when computing the F1 score, and this should ease this problem.

While REMERGE does not provide 100% accuracy and does make mistakes, its performance is satisfactory and allows for research on innovation to go beyond what is available solely through PATSTAT. Compared to existing algorithms, it is a much more flexible approach. The flexibility of the algorithm is multifaceted:

- **almost agnostic** on the company database: it does not depend on the specific company database
- **open source**, written in python and R, with documentation to maximize readability
- **expandable**: changes to the clean-up dictionaries and addition of variables is relatively easy
- **efficient**: uses the NumPy and Pandas python libraries

- **fast**: most of the resource-intensive phases of the algorithm run in parallel using iPython multicore interface
- **automatized:** almost all tasks – except the hand-labeling of the training set - do not require manual intervention. This allows, for example, re-running the algorithm as soon as an update of one of the concerned databases is available
- **no black box:** the insides of the algorithm are easily interpretable because of the readability of python code, and the fact that the model is based on the concept of running a regression.

For example, improvements in results could be obtained when matching PATSTAT to Orbis by using the ownership structure of companies. We avoided this to maintain agnosticism on the source of company information. Additional improvements could be obtained by adding more variables for the regression. For example, additional company financial information such as the expenditure in R&D helps in ascertaining the probability that a company has matching a PATSTAT entity. A list of previous company names would also be very useful: as we have seen, REMERGE would not need to standardize all names into a single one, but could work with all separately. Even better if this list was accompanied by the date at which the name was changed. These are only examples: there is no rule to determine whether some piece of information will be useful for matching. In general, anything that is useful for a human being in answering the three questions "Is this PATSTAT entity a company?", "Can this company have applied for patents?" and "Is this PATSTAT entity the same as this company?" will also be useful for our algorithm. REMERGE is a supervised machine learning algorithm and thus needs human input to learn how to weight the variables with which it can work. While human input is still central to the algorithm, the time required for remerge to work at a satisfactory level is limited.

Conclusions

In this paper we outlined the features of REMERGE, a regression-based supervised machine learning algorithm that allows to automatize the linkage of databases that are strongly different in terms of available information. We applied this algorithm to the linkage of PATSTAT with Amadeus. The algorithm has a good performance overall, while maintaining manual work to a minimum. In order keep the algorithm as general as possible, we avoided making it dependent on external sources of information such as the EEE-PPAT table. This kind of external information and other information not currently exploited in the Amadeus database can be exploited to further improve on the performance of the algorithm.

The results of the matching efforts include a confidence assessment on the matches, which should be included in research, or at a minimum can be used as a starting point for manual checks. The algorithm can be reused for the linkage of future versions of the PATSTAT database after a quick adaptation of the source code.

References

Du Plessis, M., Van Looy, B., Song, X., and Magerman, T. (2009) Data Production Methods for Harmonized Patent Indicators: Assignee sector allocation. *EUROSTAT Working Paper and Studies, Luxembourg*.

EPO/OHIM (2013) Intellectual property rights intensive industries: contribution to economic performance and employment in the European Union. *Industry-Level Analysis Report of the European Patent Office and the Office for Harmonization in the Internal Market*.

Fellegi, I. P. and Sunter, A. B. (1969). A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210.

Herzog, T. N., Scheuren, F. J., and Winkler, W. E. (2007). *Data Quality and Record Linkage Techniques*. Springer, New York.

Huberty, M., Serwaah, A., and Zachmann, G. (2013a) A flexible, scalable approach to the international patent "name game". *Bruegel Working Paper*.

Huberty, M., Serwaah, A., and Zachmann, G. (2013b) A scalable approach to emissions-innovation record linkage. *Bruegel Working Paper*.

Lanzi, E., I. Haščič and N. Johnstone (2011), Efficiency-improving fossil fuel technologies for electricity generation: Data selection and trends. *Energy Policy* 39(11):7000–7014.

Lotti, F. and Marin, G. (2013) Matching of PATSTAT applications to AIDA firms: Discussion of the Methodology and Results. *Questioni di Economia e Finanza*, 166.

Lybbert, T. J. and Zolas, N. J., Getting Patents & Economic Data to Speak to Each Other: An 'Algorithmic Links with Probabilities' Approach for Joint Analyses of Patenting & Economic Activity (2012). *WIPO Working Paper* No. 5-2012.

Thoma, G., Torrisi, S., Gambardella, A., Guellec, D., Hall, B. H., Harhoff, D. (2010) Harmonizing and Combining Large Datasets – An Application to Firm-Level Patent and Accounting Data. *NBER Working Paper* No. 15851.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B (Methodological)*, 58(1):267–288.

Appendix

1. Data from Wharton Research Data Services (WRDS): variables and manipulation

We extracted the following variables from WRDS:

- IDNR: identification variable
- CLOSDATE_year: reference year for financial data
- IFAS: intangible fixed assets
- EMPL: number of employees
- OPRE: operating revenues
- RD: research and development expenditure
- NAICS_CORE_CODE: NAICS sector
- NAME: company name
- ADDRESS: company address
- CITY, CITY_NAT, CNTRYCDE: city, city in the original language, country code
- TYPE: company legal form

We extracted data for EU27 in the years 2001, 2006, 2009, 2011, and merged all the extractions steps into a single database. If a company appears in multiple years, we keep the last non-missing values of the above mentioned variables.

We also ran other additional extractions:

- an additional extraction from the company subsidiaries table in WRDS, and we used that to calculate the number of subsidiaries of every company
- three separate extractions from the available subsets of companies (V, V+L, V+L+M subsets) in order to obtain an additional variable that refers to the size of the companies (eg. a medium sized company belongs to V+L+M but not to V+L or V).

2. Training sets and issues related to the linkage model

The linkage algorithm uses:

- one training set for the estimation of the models
- one training set for the selection of the best probability cutoff.

We treat record linkage as if it was a classification problem, i.e. for every candidate we assign an estimate of the probability that it is a true match. There are consequences to this:

- The training set is made of a list of PATSTAT entities, each of them associated to a list of candidates. This means that rows referring to the same PATSTAT entity are not independent.
- The algorithm uses L1-penalized logistic regression and thus maximizes the likelihood under a constraint for the vector of coefficients.

- The above two points imply that the algorithm implicitly optimizes the accuracy (or error rate) of the predictions.
- The estimated probability of match for every pairing is forced to zero if it is not the highest probability among the candidates for a single PATSTAT entity.
- For every PATSTAT entity, there will be at most 1 estimated true match. The number of estimated false matches will be equal to the number of candidates for that PATSTAT entity, minus one. This means that if we increase the number of candidates for a PATSTAT entity, we always increase the number of estimated false matches.
- When considering the training set with multiple candidates for a single PATSTAT entity, we do *not* want to have accuracy as our objective function, because accuracy weights true positives as much as true negatives. But true negatives are of no interest, as this relationship always holds:

$$\uparrow \textit{candidates} = \uparrow \textit{estimated false matches} = \uparrow \textit{true negatives}$$

- This is why we avoid using automatic Cross Validation, but instead we use a separate training set on which we calculate the models' F1 score (that is not influenced by the true negatives) to select the best one.