Disambiguation and Co-authorship Networks of the U.S. Patent Inventor Database (1975-2010)

Ronald Lai

Alexander D'Amour
Department of Statistics
Harvard University
Cambridge, MA 02138
adamour@iq.harvard.edu

David M. Doolin
Fung Institute for Engineering Leadership
UC Berkeley College of Engineering
Berkeley, CA 94550
doolin@berkeley.edu
ORCID: 0000-0003-2135-4660

Guan-Cheng Li
Electrical Engineering and Computer Science
UC Berkeley College of Engineering
Berkeley, CA 94550
guanchengli@eecs.berkeley.edu

Ye Sun
Grantham, Mayo, Van Otterloo & Co. LLC
40 Rowes Wharf, Boston, MA 02110
Edward.Sun@GMO.com

Vetle Torvik
Graduate School of Library and Information Science
University of Illinois at Urbana-Champaign
501 E Daniel St, Room 221
Champaign, IL 61820
vtorvik@illinois.edu
ORCID: 0000-0002-0035-1850

Amy Yu

Lee Fleming
Fung Institute for Engineering Leadership
UC Berkeley College of Engineering
Berkeley, CA 94550
lfleming@ieor.berkeley.edu
ORCID: 0000-0001-6907-1278

June 17, 2013

1

Disambiguation and co-authorship networks of the U.S. Patent Inventor Database (1975-2010)

Abstract: Research into invention, innovation policy, and technology strategy can greatly benefit from an accurate understanding of inventor careers.   The United States Patent and Trademark Office does not provide unique inventor identifiers, however, making large-scale studies challenging. Many scholars of innovation have implemented ad-hoc disambiguation methods based on string similarity thresholds and string comparison matching; such methods have been shown to be vulnerable to a number of problems that can adversely affect research results. The authors address this issue contributing 1) an application of the Author-ity disambiguation approach (Torvik, Smalheiser, et al., 2005; 2009) to the US utility patent database, 2) a new iterative blocking scheme that expands the match space of this algorithm while maintaining scalability, 3) a public posting of the algorithm and code, and 4) a public posting of the results of the algorithm in the form of a database of inventors and their associated patents. The paper provides an overview of the disambiguation method, assesses its accuracy, and calculates network measures based on co-authorship and collaboration variables. It illustrates the potential for large-scale innovation studies across time and space with visualizations of inventor mobility across the United States. The complete input and results data from the original disambiguation are available at (http://dvn.iq.harvard.edu/dvn/dv/patent); revised data described here are at (http://funglab.berkeley.edu/pub/disamb_no_postpolishing.csv); original and revised code is available at (https://github.com/funginstitute/disambiguator); visualizations of inventor mobility are at (http://funglab.berkeley.edu/mobility/).

Keywords: disambiguation, patents, networks, inventors, careers.

1. Introduction

Reasonably complete though raw United States patent data first became available in the 1990s. While these data enabled research in the fields of technology and innovation, the publication of a curated dataset by the National Bureau of Economic Research (NBER) gave a much broader set of researchers access to the patent data (Hall et al., 2001). The NBER effort made patent data accessible to a larger community of researchers that lacked the resources and hardware or programming skills to access the data. The original NBER database included inventor names, firm name and state level data but did not identify unique inventors over time.

Uniquely identifying inventors presents at least two challenges. First, the United States Patent and Trademark Office (USPTO) does not require consistent and unique identifiers for inventors. For example, an author of this paper is listed as Lee O. Fleming on patent 5,136,185 (Fleming, 1992) but as Lee Fleming on patent 5,029,133 (Fleming, 1991). Both inventors work for Hewlett Packard, both invent semiconductor circuits, and both live in Fremont, California – without personal knowledge, with what confidence could we infer that this is the same inventor? Moving directly into the second challenge, could we repeat this process for millions of inventors? Accurate and automatic disambiguation of the entire patent record requires careful algorithm design to ensure scalability and, even then, significant computational resources to ensure feasibility. For example, the brute force approach to compare all pairwise inventor-patent records is not feasible at full scale for any but the most powerful computers in existence.

In recent years there has been a flurry of activity surrounding the problem of name ambiguity in bibliographic records such as journal and conference paper collections (reviewed by Smalheiser and Torvik, 2009). Of particular note, and strong motivation for this paper, recent work has highlighted the pitfalls of poor or simplistic author disambiguation; for example: Raffo and Lhuillery (2009) demonstrate differences in econometric inferences, Diesner and Carley (2009) show differences in entity resolution and relationships in newspaper corpora, and Fegley and Torvik (2013) illustrate dramatic distortions in social networks due to non-existent or poor disambiguation. Due to space constraints, we will not make similar comparisons here, but recommend the reader to this literature, and encourage the community to heed this literature's concerns in future analyses.

1.1 Existing work and contribution

Our paper contributes 1) an application of the Author-ity disambiguation approach (Torvik, Smalheiser, et al., 2005; 2009) to the US utility patent database, 2) a new iterative blocking scheme that expands the match space of this algorithm while maintaining scalability, 3) a public posting of the algorithm and code, and 4) a public posting of the results of the algorithm in the form of a database of inventors and their associated patents. The work builds directly on prior efforts by a variety of innovation researchers (Fleming and Juda 2004; Singh 2005; Trajtenberg

et al., 2006; Raffo and Lhuillery 2009; Carayol and Cassi, 2009; Lai et al., 2009; Pezzoni et al., 2012). The database provides unique identifiers for each patent's inventors across the entire time period. It also provides social network measures by each inventor, by three-year periods from 1975 to 2010. To illustrate applications of the data, we provide movies of inventor mobility across large U.S. states since 1975. The algorithms and code are made public to encourage further development and improvement by the community of patent and innovation investigators. In addition to improved disambiguation, the Harvard Dataverse Network (DVN) and GitHub websites provide a network interface that enables a researcher to subset the co-authorship networks of inventors. [1] Output formats support both regression analysis and graphical network programs.

1.2 Precís

The second section of the paper ("*Overview of dataset preparation*") provides an explanation on how the inventor dataset is created; the third section ("*Disambiguation: overview, theory, and implementation*") provides a non-technical overview and explanation of the disambiguation processes; the fourth section ("*Results and accuracy metrics*"), describes how we report results and accuracy; the fifth section ("*Disambiguated data and illustrative applications*") illustrates applications of the data. Appendices include patent data descriptions, listing of data and results distributed through the Harvard Dataverse Network and schemas used in and produced by the disambiguation.

---

[1] Original data are stored at http://dvn.iq.harvard.edu/dvn/dv/patent. More recent disambiguation code and updated data are available at Fung Institute and GitHub websites:
https://GitHub.com/funginstitute/downloads and
http://funglab.berkeley.edu/pub/disamb_no_postpolishing.csv.
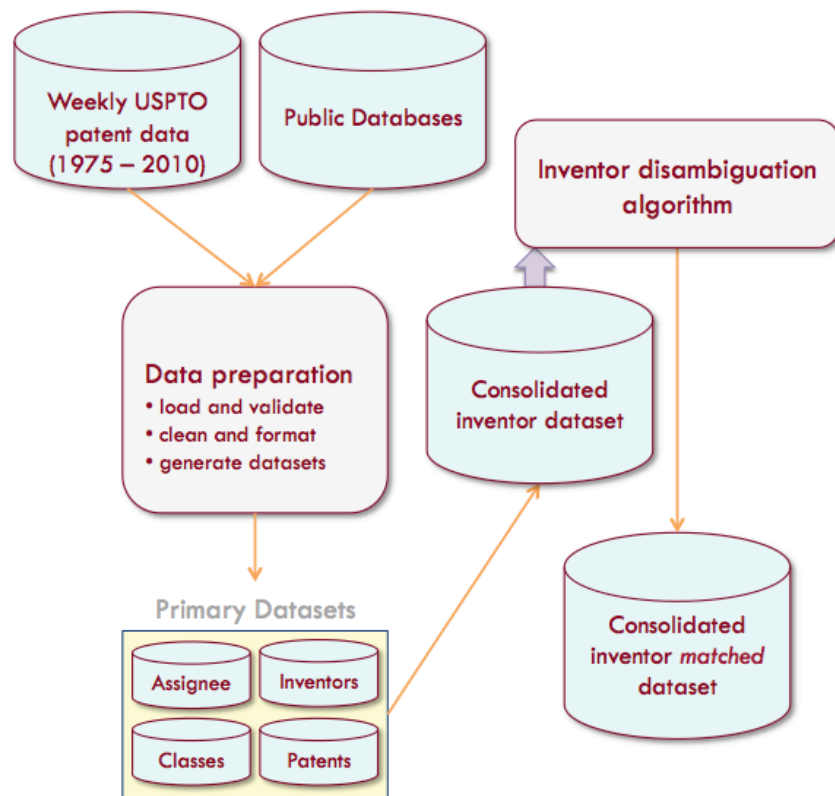
2. Overview of dataset preparation

Figure 1: The patent disambiguation process. Squares represent code and processing; canisters represent databases.

Figure 1 illustrates an overview of the patent disambiguation process. Source data come mainly from the NBER database (Hall et al., 2001) and directly from the US Patent and Trademark Office (USPTO) weekly publications.[2] The data preparation step generates four primary datasets containing the relevant inventor, patent, assignee, and classification data that will be used in the inventor disambiguation process. These datasets are consolidated and then fed into the disambiguation algorithm that outputs the inventor career database.

Dataset preparation consists of obtaining patent and patent-related data from primary and secondary sources, parsing raw patent data as needed, cleaning both patent and secondary source

---

[2] Some of the early NBER data are missing and are supplemented by the 1998 Micropatent CD product (http://www.micropat.com/static/index.htm). We would like to acknowledge the donation of these data from Corey Billington and Ellen King of Hewlett-Packard. This completes approximately 70,000 gaps in data for records from 1975-1978.

data, and consolidating the parsed and cleaned inventor and patent data into a single database containing inventor-patent instances, or records, to be disambiguated.

2.1 Primary data sources

The final inventor, assignee, inventor, patent and classes datasets were built using primary data sources from the USPTO and the NBER. The USPTO makes up-to-date patent data available on their public web resource[3] through collaborations with the European and Asian patent offices. The weekly data file is a concatenated list of granted patents, where each patent is represented by an XML document (that is, all files are merged chronologically). The NBER patent database contains patents granted from 1975-1999 and is publicly available[4]. Since the patent office only began automating data storage in 1975[5], we are utilizing information from 1975 onwards. To the best of our knowledge, there is no freely available and comprehensive computer database containing U.S. inventor information before 1975, though bulk download of images and OCR text (of variable quality) files are available. [6]

2.2 Secondary data sources

In addition to the primary data sources, we merged in data from secondary public data sources to help identify inventors. These secondary data sources include the USPTO CASSIS dataset[7], the

---

[3] USPTO provides weekly Bibliographic Information for Patent grants through its Sales Order Management System (SOMS) Catalog. https://EIPweb.uspto.gov/SOMS

[4] See Hall et al., 2001 at http://www.nber.org/patents/.

[5] NBER provides limited data from 1963-1999 but only provides inventor data from 1975-1999. Since inventor information is necessary in our disambiguation algorithms, we have only matched inventors to patents granted after 1975. Further information about the inventor dataset can be found at: http://www.nber.org/patents/inventor.txt.
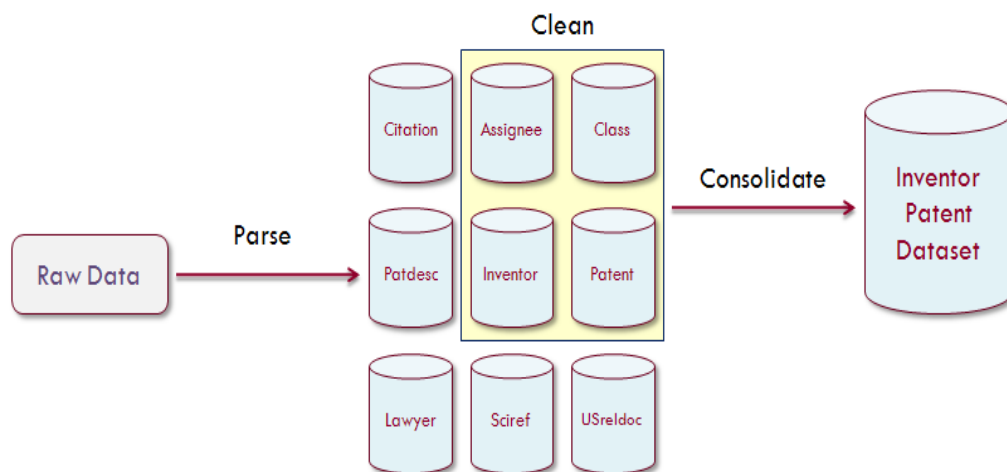
[6] Google Books: http://www.google.com/googlebooks/uspto-patents.html.

[7] Patents CLASS: Current Classifications of US Patent Grant Publications 1790 to Present' (Code: EIP-2050P-DD): http://www.uspto.gov/web/offices/ac/ido/oeip/catalog/products/pp-o2w-3.htm#classP2050dd

National Geospatial-Intelligence Agency country files[8], the US Board on Geographic Names[9] and NBER File of Patent Assignees. [10]

When a patent is granted, the USPTO assigns multiple alphanumeric codes to classify the technology. As technology advances, the USPTO creates new classifications and updates previously coded patents. These classification changes are indicated in CASSIS, a dataset that is updated bimonthly. Classifications reflect the November 2009 concordance. Geographic metrics are sourced from public databases such as the National Geospatial-Intelligence Agency and the US Board on Geographic Names, current through 2009 (recent efforts have improved upon this, see Johnson 2013). Since assignees are often public firms, we leverage the NBER Patent Data Project (PDP). [11] Through a combination of the NBER PDP data and heuristic string matching procedures, we have incorporated NBER's unique assignee identifier, PDPASS, into our input dataset. [12]

2.3 Preparing the inventor dataset

Figure 2: Preparing patent data for disambiguation.

Figure 2 provides a schematic of the data preparation process. The initial step parses the raw data input. In order to minimize redundancy, several smaller tables were created and joined together using unique patent and inventor identifiers rather than generating one large dataset containing all unique combinations of patent information. USPTO patent data contain 60+ fields of information. If we were to restrain our data into one primary dataset, unique permutations of each field would be difficult to manage.

The smaller, independent datasets consist of assignees, citations, patent technology classes, inventors and patents. The data within the independent datasets are further cleaned before being consolidated for disambiguation. Cleaning includes removing excess whitespace, standardizing date formats and similar tasks. Consolidation includes adding location and assignee data, which are matched between the primary and secondary data sources to merge longitude, latitude, and assignee identifier information within the inventor and patent datasets. The cleaned, consolidated data comprise the input dataset for the disambiguation process.

3. Disambiguation: overview, theory, and implementation

Put simply, the challenge in studying inventor careers from the raw patent data is determining which patents belong to the same inventor career. The patent data include unique identifiers for each patent, but not for inventors, so clustering of patents by distinct inventors on a large scale requires a procedure that can cluster together patents by the same inventor and distinguish them from patents by other inventors with the same or similar name. The process of clustering together likely distinct inventors is called disambiguation.

Many existing disambiguation algorithms cluster records by calculating similarities between pairs of records, then grouping together sets of records that exceed arbitrary thresholds, or by assigning ad hoc weights to record attributes such as inventor name, assignee, technology class, co-inventors, etc. in order to determine a unitless match score (Fleming and Juda, 2004; Singh 2005; Trajtenberg et al., 2006; Lai et al., 2009). Above a predetermined threshold, two records would be declared a match. This weighting scheme is then often tuned to optimize results with a hand-curated dataset.

However, manually optimizing a disambiguation scheme is susceptible to a number of problems that our machine learning approach mitigates. The first is model-dependence; the linearly weighted combination fails to capture clear and non-trivial interactions between certain feature similarities. For example, if two records match on assignee, but the assignee is large and has patents in multiple fields, the technology class overlap can have a large impact on how the

assignee match ought to be considered (for a small firm, for whom all patents are in the same technology, class overlap may add little information, but for a large firm, it may add a great deal). Linear specifications could handle these dependencies by introducing interaction terms, but this model-selection problem would be cumbersome and lead to non-linearities whose predictive accuracy would be hard to assess.

The second problem with manual optimization is that the dataset being used to train the weights (that is, the dataset used to inform the selection of the weights), no matter how accurate, typically represents a small, biased sample. Inventors in these gold-standard datasets tend to belong to the same communities (e.g. the BIIS dataset in Trajtenberg et al., 2006, or our dataset, based on the Marschke survey of academic inventors, see Gu et al., 2008) or tend to be more prolific than average, making them more visible to researchers doing a manual survey. Despite the best efforts by researchers, hand-curated datasets are often incomplete; many inventors do not maintain a complete and updated (let alone published) list of patents that they have invented. Even carefully sampled and executed surveys remain vulnerable to bias, for example, some inventors remain difficult to contact (e g., the deceased). Hand curated datasets can be a poor choice for training if the biases in the data outweigh the benefits of true data. Unbiased datasets or those with bias that doesn't affect the goals of the analysis can also be extremely difficult and costly to create.

Finally, valuable information is lost when assigning each pair of records a unitless match score, rather than a probability having a natural interpretive value. Determining such match scores requires judgment and domain-specific experience on the investigator's part. In contrast, probabilities can be estimated by measuring the statistical properties of the data.

Following the work on PubMed by Torvik et al., (2005), and Torvik and Smalheiser (2009), and on patents by Carayol and Cassi (2009) we avoid ad hoc decisions and mitigate these limitations by:

1) Training a probabilistic model that a) assumes only multidimensional order and therefore intrinsically captures non-linear and interaction effects among the predictive features, b) allows for correcting transitivity violations among triples of inventor-patent instances based on principles of probability theory, c) provides a natural likelihood-based framework for clustering.

2) Training with large, diverse, and automatically generated training sets of highly probable matches and non-matches sampled across the entire dataset so that selection bias, training variance, and manual effort is reduced.

3) Using intentionally generic predictive features so that the trained model can be applied to new data.

Technical details can be found in the references, and we encourage the interested reader to consult them. Our intent here is to broadly characterize the model and algorithm to a non-technical audience, so that innovation scholars might make more informed and effective use of the disambiguated data.

3.1 Overview of terms and a process flow diagram

The raw data in our disambiguation are not patents per sé, but patent authorships, or what we call *inventor-patent instances.* Each instance corresponds to a name appearing on a patent – for example, a patent with three authors contributes three inventor-patent instances. The core of the disambiguation algorithm is to consider all pairs of these inventor-patent instances and to determine whether or not they belong to the same inventor career. The primary unit of analysis in the core algorithm is therefore pairs of inventor-patent instances, also known as *inventor-patent pairs* or co-authorship pairs. The broader descriptive term in the literature is a "*record pair*"; we use the terms interchangeably.

Viewed in this way, the disambiguation problem boils down to a *classification* problem, where we wish to label inventor-patent pairs as *matches* – that is, pairs where both inventor-patent instances come from the same career – or *non-matches*. Classification is one of the fundamental problems in statistical machine learning, and has received wide treatment (See for example *Elements of Statistical Learning* by Hastie et al or *Pattern Recognition and Machine Learning* by Bishop). A classification algorithm or *classifier* takes in a set of attributes or measurements associated with an object and, based on a set of previously "*learned*" representative examples, uses these attributes to label the object with a class. In this case, the objects are inventor-patent pairs; the attributes are similarity scores obtained by comparing the entries associated with each inventor-patent instance in the pair, for example the similarity in their names, or the distance between their addresses; and the class is either match or non-match. Once this classification has been performed, inventor careers can be constructed by iteratively clustering together patent pairs that are determined to match.

The general Author-ity approach lends itself to a choice of classifier for estimating the matching odds. Two such classifiers are created; one for each of the two sets of attributes assumed independent as a part of the process of generating the training sets automatically. Torvik et al. (2005) used what can be described as multidimensional isotonic regression, which enforces ordering constraints on the attributes' contribution to the match odds. This captures non-linearity and combinatorial effects of the attributes, at least within each of the two sets of attributes. The two isotonic regression odds functions are then combined with a method for calculating a prior probability of match using a Bayesian formula in order to calculate an actual match probability.

We apply a slightly different implementation of a technique known as a Naïve Bayes Classifier to classify inventor-patent pairs as matches or non-matches. The classifier is similar to Naïve

Bayes in that it relies on seemingly naïve assumptions of independence between certain sets of attributes, and it uses Bayes rule to convert the likelihood that an object of a particular class has a particular set of attributes (learned from the example sets) into the posterior probability that an object with a particular set of attributes belongs to a particular class. Despite its simplicity, Naïve Bayes classifiers perform surprisingly well in real problems (Hastie et al 2001, Bishop 2006, Lewis 1998, Rish 2001, Zhang 2004); the Author-ity approach differs in that it does not require independence between all attributes, only between the sets of attributes.

The general procedure proceeds as follows. We begin by defining a representation of the attributes of an inventor-patent pair that we call a *similarity profile* (essentially selecting a subset of characteristics to compare and defining how to compare them). Once we have settled on a representation, we obtain training sets, or sets of inventor-patent pairs whose labels are assumed "known". We generate these training sets automatically using a procedure that is subject to assumptions of independence between different parts of the similarity profile. Using these training sets, we learn the likelihood that matching pairs and non-matching pairs could give rise to each similarity profile. We then compute similarity profiles for inventor-patent pairs in the larger database, and use the likelihood values to determine whether there is enough evidence to declare a pair of inventor-patent instances a match. Finally, we resolve any conflicts that arise in the match and non-match classifications between different inventor-patent pairs, and convert these into clusters of inventor-patent instances that represent full inventor careers.

Unfortunately, the computational cost of examining every inventor-patent pair in the database is prohibitive. To reduce computational effort, we apply a heuristic by *blocking* the records by predetermined criteria that are likely to be satisfied by most matching record pairs, such as matching exactly on first and last names. We only compute similarities between pairs of records within these blocks (rather than the entire database). Using the likelihood values associated with the computed similarity profiles, we then iteratively develop working clusters of each inventor's patents within each block. Repeated rounds of agglomerative clustering terminate when the log-likelihood of the clustering solution hits its maximum. To avoid confusion, disambiguation refers to the entire "who's who" process, while matching refers to the direct comparison of records to determine unique inventors.

Torvik and Smalheiser pioneered this semi-supervised classification approach to the disambiguation problem. We refer to the whole procedure, including similarity profile construction, automatic training set generation, technical constraints on the training likelihoods, and simple blocking heuristics employed to reduce computation as the *Author-ity approach*. Our contributions are an iterative blocking scheme (defined in Sections 3.4 and 3.5), and the application of this algorithm to the US patent record. Figure 3 illustrates our disambiguation process. Each of the blocks is explained in detail below.
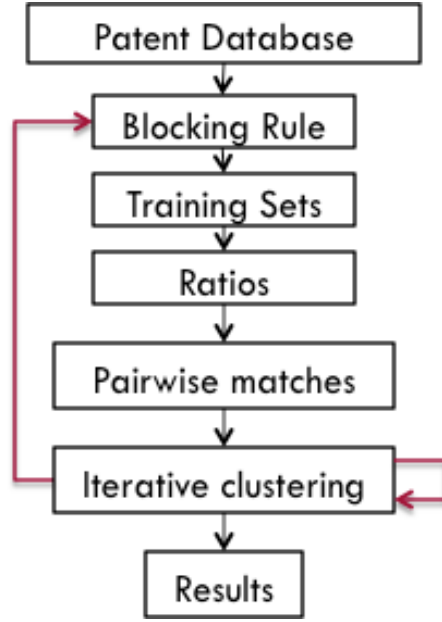
Figure 3: Steps in the iterative disambiguation process.

3.2 A Bayes classifier for disambiguation

The core statistical idea in the Author-ity approach is the application of Bayes' theorem to derive the probability that an inventor-patent pair is a match given its similarity profile (the Author-ity approach also works on author-paper record pairs). Formally, if we define M to be the event that an inventor-patent pair is a match and N to be the event that it is a non-match, we can use Bayes' theorem to write the probability of M given that we observed a similarity profile x as

$$P(M|x) = \frac{P(x|M)P(M)}{P(x|M)P(M) + P(x|N)(1 - P(M))}. \quad (1)$$

Here, $P(M|x)$, the *posterior probability of a match*, is the quantity of interest. $P(x|M)$ and $P(x|N)$ are the *likelihoods of the similarity profile* given that the pair is a match or a non-match, respectively. $P(M)$ is the *probability of a match*, and must be specified by the user (we set this, based on simple baseline probabilities within each block, described below).

It is often easier to work with the *posterior odds of a match*, which has a one-to-one relationship with the posterior probability:

$$\frac{P(M|x)}{1 - P(M|x)} = \frac{P(M)}{1 - P(M)} \frac{P(x|M)}{P(x|N)}. \quad (2)$$

The key factor here is the second fraction on the right-hand side that is called the *likelihood ratio of a match*, and quantifies the evidence for a match versus a non-match. Intuitively, the posterior odds of a match are the prior odds multiplied by this likelihood ratio. We call the likelihood ratio the *r-value*, defined as

$$r(x) = \frac{P(x|M)}{P(x|N)}. \tag{3}$$

The *r*-value is determined directly from the training set, simply by calculating the proportion of times that the similarity profile x appeared in the match set and the non-match set and taking their ratio. To account for noise that can arise from rare similarity profiles, we modify these raw values slightly to enforce monotonicity constraints and to interpolate or extrapolate missing *r*-values (Torvik et al. 2005), a procedure discussed in the section on training sets (Section 3.6).

Eq. 2 is easily converted back into an expression for $P(M|x)$, so that we can write the quantity of interest in terms of the r-value and prior information:

$$P(M|x) = \frac{1}{1 + \frac{1 - P(M)}{P(M)}\frac{1}{r(x)}}. \tag{4}$$

The prior probability of a match $P(M)$ is specified *a priori* based on, for example, the size of the block under consideration (for example, a larger block makes a match less likely). We discuss prior probabilities in Section 3.7.

Next, we provide a detailed but less technical overview of each aspect of the disambiguation algorithm in turn, providing citations when appropriate for readers interested in technical detail.

3.3 Inventor-patent instance data

The unit of analysis in our disambiguation is an *inventor-patent instance*, also referred to as an *inventorship*, each corresponding to a record in the input dataset. Each record contains *attributes* used for disambiguation, such as the inventor's first and last names, the latitude and longitude of the inventor's home town, the patent assignee, and others as explained below. Each inventor-patent instance occurs only once. In contrast, a patent may appear multiple times, once for each inventor listed on the patent. For example, disambiguating a patent with four inventors would result in four inventor-patent instances, hence four records. The input dataset was created by merging data from the relevant databases to create a table containing over 8 million inventor-patent instances.

An inventor's name is the most distinguishing attribute. In the raw dataset, the inventor name is split into first name (with middle name, when present) and last name (with suffix, when present).

We define a full name as having both first and last name present, which is available for 99.99% of records. Having a full name for disambiguating patent inventors is a major advantage over disambiguation of journal and conference paper collections, which often lack authors' first names.

When the USPTO issues a patent, the original owner of the patent, which is most often the employer of the inventor (Singh and Fleming, 2010), is listed as the *assignee* on the patent grant. The assignee's name would ideally be enough to identify the firm that holds the patent, but problems arise from misspellings, from different forms of the same company's name and from subsidiaries having completely different names from the parent. For example, consider "IBM" versus "International Business Machines", the same assignee with different forms of the company name.

A combination of address features (city, state, and country) was matched against public geographic databases from the National Geo-spatial Intelligence Agency to extract longitude and latitude of inventor's physical location. Using geographic coordinates permits calculating distances between inventors where simple address string similarity does not accurately capture the "closeness" of two different addresses. Street addresses were not available for all records and were not used. Converting the variety of geographical data fields to a longitude and latitude makes the comparison more robust to missing data.

Each patent also has a list of technology classes and co-authors. These categories provide information about the inventor's area of expertise and co-authorship network, respectively. For simplicity and computational efficiency, shared co-inventor and technology classes are truncated at the first four primary classes and six co-inventors. Table 1 illustrates the fundamental challenge of disambiguation: given the attributes, are these patents invented by the same person? Further examples will develop how we answer this question in more detail.

ID	Patent	Firstname	Middlename	Lastname	Coauthor	Class	Assignee
P1	7285077	Matthew	NONE	Marx	NONE	482	NONE
P2	7321856	Matthew	Talin	Marx	MANY	704/379	Microsoft
P3	5995928	Matthew	T	Marx	ONE	704	Speechworks

Table 1 Example patents for inventors named Matthew Marx.

3.4 Blocking

In principle, we would like to classify each inventor-patent pair in the database as a match or non-match. However, exhaustive pairwise comparison requires quadratic run time. Because each inventor-patent instance must be compared with every other inventor patent instance, an exhaustive comparison of every pair of 8 million records in the patent database would require over 32 trillion comparisons, making the full problem computationally infeasible. One popular approach for making classifications based on pairwise computation feasible is blocking the records first, and restricting comparisons within blocks (On et al., 2006, Bilenko et al., 2006, Herzog et al., 2007, Smalheiser and Torvik 2009, Reuter and Cimiano, 2012).

We apply this blocking heuristic by partitioning all inventor-patent instances into groups of records that are likely to contain all true matches. This partition is defined by crudely summarizing all inventor-patent instances with a block identifier, and then bucketing all records with the same block identifier together. For example, one might block by complete first and last names, resulting in blocking identifiers like of SMITH, JOHN.

Choosing the feature set for any particular blocking scheme is a difficult balancing act. On one hand, creating blocks that are too big does little to reduce the quadratic run time. On the other hand creating blocks that are too small can rule out true matches by assigning patents from the same inventor to different blocks, creating "splitting" errors (see Section 4). To deal with this time/accuracy trade-off, we developed a novel blocking scheme that iteratively refines the blocking rules.

In our iterative blocking scheme, early iterations define fine-grained blocking identifiers, like the complete first and last-name identifier referenced above. Once we have computed working clusters of inventorships based on this blocking, we reduce the effective size of the database by collapsing clustered inventorships together, making coarser blocking schemes feasible. Technical details of this consolidation scheme are provided in Section 3.8.

For example, having applied the full-name identifier in an early blocking iteration, a later blocking iteration may define a more permissive block identifier consisting of truncated parts of first and last names, e.g., SMITH, J. This iterative scheme allows us to scale and explore a larger set of potential matches than most feasible single-blocking schemes would allow. Thus, it can catch matches that might have been missed by a more restrictive blocking.

The purpose of the iterative blocking algorithm is to expand on the potential matches captured beyond what a single rule could (e.g., only matching on exact last name and first name initial would fail to capture simple, common last name variants). As the set of potential matches expands, the prior probability of a match will decrease, making matches harder to capture (e.g., allowing a single-edit match on last name will permit Cohen and Chen to match but at a much

lower probability than within their respective blocks). The end result of our algorithm is a list of hand-crafted rules, similar to ones produced by adaptive blocking algorithms such as Bilenko et al. (2006), but here much more controlled. This decreases "surprising" results caused by idiosyncratic rules that highly non-linear machine learning algorithms can produce. Our iterative blocking algorithm also couples the training of the full similarity profile using a semi-supervised paradigm (based on the partition of attributes into largely independent sets) with the iterative blocking rules.

While we found our iterative blocking scheme to achieve a desirable compromise between scalability and efficiency, it remains an imperfect heuristic. Because each subsequent run relies on consolidated records that were collapsed according to the previous run's clustering, "lumping" errors that erroneously group together inventorships from different inventor careers can cascade from run to run. Thus, for a fixed threshold to collapse records, the iterative blocking scheme tends to cluster together records more easily than the ideal, exhaustive comparison scheme would. This bias can be decreased, however, by using a more sophisticated thresholding scheme (see Section 3.8). In the end, we found that the risk of lumping is outweighed by the decreased risk of splitting and the scalability gains that our iterative scheme provides. Other scholars, however, could decide differently, based on their substantive research question.

3.5 Pairwise comparison within each similarity profile

Classifying inventor-patent pairs requires defining a comparison function C that takes two sets of record entries and returns an n-dimensional similarity vector (or profile) $x = (x_1, x_2, \ldots, x_{n,})$ between inventor-patent instances. Each feature x_i of the similarity profile is a positive integer resulting from comparing two records, with higher values corresponding to greater similarity between respective features. Our current feature set includes: first name, middle initial, last name, inventor location, assignee, number of shared technology classes, and number of shared co-inventors.

For our example, consider the following comparison function C defined on seven features, where each feature-wise comparison returns a value in the indicated range:

1. First name [0..4]: value 0 when names are completely different, value 4 when lexicographically identical, with intermediate values determined by degree of similarity between the names being compared.
2. Middle name [0..3]: handled similarly to first names, using an appropriate comparison function to account for presence or lack of presence of a middle name.
3. Last name [0..5]: handled similarly to first and middle names, with more nuanced treatment of the last name in terms of comparison.
4. Coauthor [0..6]: number of common coauthors, where more than 6 common coauthors is set to a maximum value of 6.

5. Technology class [0..4]: values from 0 to 4 representing the number of shared technology classes between the two records being compared, where 4 is defined as the maximum feature value when four classes are in common between the records. [13]

6. Assignee [0..6]: The assignee feature incorporates both the assignee name and the assignee number, when available. Value 0 when both name and number are available and different; value 1 when one or both of the records are missing assignee information. Values from 2-5 report similarity in name, with value 6 indicating an exact match on an assignment number. [14]

7. Location [0..5]: 0 when inventors not in the same country; for inventors in the same country, values ranging from 1 to 5 are determined from distance computed from latitude and longitude (for an understanding of the data, locations which can be inferred from a US patent, and estimated errors, please see Johnson, 2013).

We can use this function to construct the similarity vectors for the inventor-patent instances containing the name "Matthew Marx" (from Table 1). The pairwise comparison of each row of Table 1 results in the following similarity vectors:

$$C(P_1, P_2) = (4,1,5,0,0,1). \qquad\qquad (5a)$$

$$C(P_1, P_3) = (4,1,5,0,0,1). \qquad\qquad (5b)$$

$$C(P_2, P_3) = (4,3,5,0,1,0). \qquad\qquad (5c)$$

The composition of the similarity profile depends on the classifier chosen for a particular round. See, for example, disambiguation Round 3 in Table 3 below, where all of the above features are used for the classifier, versus disambiguation Round 2 where the location is incorporated instead of the technology class. Regardless of the composition of the similarity vector, the core task remains mapping these profiles to the probability of a match (discussed below).

3.6 Training sets

The key hurdle in converting the disambiguation problem into a classification problem is obtaining training sets to estimate the likelihood ratios corresponding to each similarity profile.

---

[13] The USPTO organizes technology into approximately 400 classes and over 100,000 subclasses. Each patent is typically assigned to multiple classes. For more detail, please see http://www.uspto.gov/patents/resources/classification/index.jsp.

[14] The assignee is the owner of the patent. It is typically a firm, and less often an individual, university, or government (Singh and Fleming, 2010). The assignee number is given by the USPTO. For additional details, and details on all searchable fields of a US patent, please see http://www.uspto.gov/patft/help/helpflds.htm#Assignee_Name.

To obtain precise estimates, the training sets must be large, and to control the bias of the estimated ratios, the training sets must be representative. In standard classification problems, it is assumed that one has access to a large, representative set of objects whose class is known with certainty, e.g., one constructed by random sampling and manual verification. However, in the case of disambiguation, such an exact training set is difficult, and potentially impossible, to obtain. Because most pairs of patents do not match, sampling a set of pairs such that the subset of matching pairs is large enough to compute precise likelihood ratio estimates would require enormous computational effort, and manually verifying the status of the sampled pairs would be prohibitively labor-intensive (and often impossible, due to the difficulty of finding and gaining cooperation from all sampled inventors in identifying their patents – for example, deceased inventors).

To overcome this problem, we take the approach of Torvik et al. (2005) to automatically construct approximate training sets, where the pairs included in these sets are not known to be matches or non-matches with certainty, but are suspected of being so with high probability based on simple criteria. This relaxed requirement makes the construction of large match and non-match sets feasible, though this efficiency comes at a cost. [15]

For the training sets constructed in this manner to be representative, the method of selecting examples for the training sets cannot perturb the distribution of attributes in the training set. For example, a representative "match" training set should have the same distribution of similarity profiles as the set of true matches in the full database. As such, some assumptions must be made about the dependence between the features of inventorship pairs and the criteria used to select highly probable matches and non-matches. The approach we take relies on the assumption that certain parts of the similarity profiles are probabilistically independent in the true match and non-match sets. If this independence assumption holds, then restricting one subset of these features does not change the distribution of the other subset. This allows us to use one subset of features to identify highly probable matches and non-matches, while using the other subset to train the classifier. Note that these assumptions are not quite as demanding as a true Naïve Bayes classifier, where all attributes should be independent of one another.

To implement this approach, we divide the set of inventorship pair features into two mutually exclusive subsets that are assumed to be independent in the true match and non-match sets: name feature similarities (first name, middle initials, and last name) and patent feature similarities (inventor home town, patent assignee, technology class, and co-inventors). To generate a set of highly probable matches for the study of name features, we selected pairs of records that shared two or more co-inventor names and two or more common technology classifications of the

---

patents. This was done within blocks, implicitly adding an additional criterion. Similarly, to generate a set of highly probable matches for the study of patent features, we selected pairs of records where the inventor name was rare and matched exactly. We followed analogous procedures to create non-match training sets. Table 2 summarizes conditions for generating training sets.

We find the assumption of independence between name similarities and patent similarities to be rather mild, but can construct scenarios that violate the assumption. For example, an inventor who works in multiple fields may include his middle initial on patents that he files in one technology class but leave it out on patents he files in another. In such cases, the estimated likelihood ratios will incur some bias – in this example, the dissimilarities would be effectively overcounted, giving a pair of this inventor's patents that occur in different technology classes a lower likelihood ratio of matching. However, because the algorithm for constructing the training set is made explicit, for a given scenario the direction of such bias is easy to determine, and if such scenarios are too common to be tolerated, the training set algorithm can be modified. [16] To handle this particular challenge, for example, the investigator may instead choose to leave both middle initial and technology class out of the training set definitions, relaxing the condition for unbiasedness to be that the rest of the name features are independent of the patent features, and that the rest of the patent features are independent of the name features.

For the large and general-purpose disambiguation, we judged any potential bias to be worth the gains. However, it should be noted that in a number of cases where we faced difficulty, for example among inventors with East Asian names and corporate affiliations, violations of this independence assumption are more likely to be present. Future work should attempt to further improve the learning stage, for example, by incorporating same town or same assignee.

---

[16] Efforts are currently underway to make this experimentation easier to do, even for a researcher who cannot modify the original code.

	Match Set	Non-Match Set
Learn patent|name attributes	**Pairs of matched full inventor names defined as rare with respect to all inventor names.**	**Pairs of non-matching full inventor names chosen from rare name list.**
Learn name|patent attributes	**Pairs sharing 2 or more common coauthors and technology classes.**	**Pairs of inventors from the same patent.**

Table 2: Description of training sets, defining how record pairs were selected, and which feature sets they were intended to train. Learn "patent|name" attributes means to train for patent match or non-match, conditional on name attributes.

The relative frequency with which a similarity profile appears in both match and non-match training sets is used to calculate its *r*-value (Eq. 3), which is then stored in a lookup table. Note that because we compute an r-value for the whole vector, rather than a one-dimensional summary of that vector, this classification method naturally captures higher dimensional interactions between elements of the similarity profile in determining the likelihood of a match.

Because they are estimated quantities, the raw *r-values* can be noisy, and need to be smoothed, though smoothing requires some assumptions. One reasonable assumption is that inventor-patent pairs with greater similarity ought to have greater match probability, however this can be violated if certain similarity profiles are rare. To remedy this problem, we follow Torvik et al. (2005), and define a *product order* between similarity profiles x and y where we say x is greater than y if and only if every entry of x is greater than or equal to every entry of y, or formally, $x \leqslant y \Leftrightarrow x_i \leq y_i \ \forall \ i = 1, 2, \dots n$, where n is the dimension of the similarity profile. We use this ordering to explicitly impose a *monotonicity constraint*, such that for any two similarity profiles x and y, if $x \leqslant y$ then $P(M|x) \leq P(M|y)$. It can be shown that this is equivalent to imposing monotonicity on *r*-values: $P(M|x) \leq P(M|y) \Rightarrow r(x) \leq r(y)$.

When profile A is greater than profile B, each element in A is equal to or greater than the corresponding element of profile B, and A must map to a higher match probability than B. Consider the similarity profiles (Eqs. 5a, 5b, 5c) constructed from Table 1 using inventor name "Matthew Marx." Let $A = (4,3,5,0,1,0)$ and $B = (4,1,5,0,0,0)$. Comparing element-wise, $i = 1, 2, \dots, 6$; $a_i \in A, b_i \in B, a_i \supseteq b_i$, thus $A \supseteq B$. Using *r*-values obtained from the actual disambiguation, for profile A, $r = 0.593733$, and for profile B, $r = 0.000472872$. (As it turns out, similarity profile A indeed does reflect the same individual, and similarity profile B does not.)

We use the monotonic ordering assumption to smooth the *r*-values that are observed in the training set and to interpolate or extrapolate when new similarity profiles that did not appear in

the training set are encountered in the larger database. We perform this smoothing by finding the set of monotonic *r*-values that has the minimum weighted squared distance from the raw *r*-values, where the weights are proportional to the number of times the corresponding similarity profile appeared in the training sets. This optimization problem can be solved using quadratic programming (Torvik et al., 2005).

Unfortunately, a small or zero value in the denominator can greatly influence the *r*-value. In order to dampen the influence of extreme ratios, we apply a Laplace correction (Hazewinkel 2001) equal to 5, following Torvik et al.'s (2005) experience in disambiguating the similarly sized Medline data. Comparing the numbers in between a typical vs. outlier influence on *r*-values indicated ~100 similarity profiles that required a Laplace correction.

Training sets, whether based on inventor names, technology class, co-inventor or the like depend strongly upon the particular blocking rule. Hence, after blocking and before each round of disambiguation, training sets are recreated and a new *r*-value lookup table is built, specific to each round of blocking.

3.7 Prior probabilities

The prior match probabilities *P*(*M*) for pairs within each block are determined in two steps. In blocking rounds after the first, when working clusters have been defined previously, we use the ratio of within-cluster pairs in a block to the total number of pairs in that block to compute an initial value for *P(M)*. The initial blocking round starts each cluster in the block with only one record and computes the same ratio (essentially the inverse of the number of pairs in the block, assuming no pre-consolidation for exactly similar fields, as described below).

We then adjust this initial prior probability for each block according to the frequency of each part of its block identifier, *i.e.*, it is penalized if and only if all parts of the block are both very common; otherwise, it gets augmented for each part of the block identifier. In our current engine, the factor of modification is the logarithm of the ratio of the maximum occurrence of a block identifier to the occurrence of the current block identifier. In other words, the prior probability decreases, when identifiers are common, because greater skepticism of a match is warranted.

3.8 Inventor-patent pair matching and iterative clustering into careers

Given a mapping from every possible similarity profile to its likelihood ratio *r*, calculating the probability that any two inventor-patent pairs match becomes relatively simple. Before comparing the two records, the prior match probability *P*(*M*) is calculated based on the type of blocking that was performed. The two records are compared field-wise to generate a similarity profile. The probability of a match, given an observed similarity profile and prior probability, is then calculated from Equation 4.

These pairwise probabilities must then be grouped by inventor, in order to collect all the patents in each career. We accomplish this grouping with repeated iterations of working or potential clusters (before the final cluster, a cluster is technically "working" or "potential"). A cluster consists of 1) the inventor's patents, 2) a *cohesion* value, and 3) a *cluster representative record*. Cohesion is the arithmetic average of some of the pairwise comparison probabilities among the members. The cluster representative record has the most attributes in common with all the records in the cluster.

The iterative clustering process follows each round's matching. In the very first round of blocking, working clusters begin at most as the individual inventor-patent pairs (with the exception of the pre-processing, described below). In subsequent rounds, working clusters begin based on the previous round's last clusters. First, a similarity profile is computed between cluster representatives, followed by the *r*-value lookup for the similarity profile, after which the final match probability of the two representatives is calculated. If the match probability of the representatives does not pass a minimum threshold (empirically set at 0.3 to minimize run time, based on the observation that no final clustering ever occurred beneath that), it is assumed that the clusters are not of the same inventors and that running the full comparison process would be a waste of time. This prescreening step can significantly accelerate the overall disambiguation process.

If the comparison between working cluster representatives passes the minimum threshold, exhaustive comparisons between members of the two clusters are performed, along with an effective comparison count based on the size of the two clusters. The introduction of the effective comparison count is to allow clusters representing inventors of high mobility to merge. Instead of having to meet the requirement that the average of all comparisons between members in the two clusters surpasses a certain threshold, the two clusters need only to pass the threshold for the average of the maximum effective comparison count number of probability values among all the exhaustive comparisons. If the effective comparison count average is greater than the threshold, the two clusters will merge, and the cohesion value of the new cluster is set to the effective comparison count average, after which a new representative can be determined.

A sequence of monotonically decreasing thresholds is set, with the expectation that more similar clusters should agglomerate first. If the comparison of two working clusters yields a probability greater than a given threshold, the two clusters will consolidate into a larger working cluster, and the within-cluster density and cluster representative are updated. The iterative grouping within a block starts again with a lower threshold if no more working cluster representative pairs qualify for consolidation under the current threshold. The loop continues until all thresholds are passed, signaling the end of the disambiguation of the block based on its current blocking mechanism. These working clusters are then fed into the next round with different blocking rules and

possibly different similarity profiles. The working clusters at the end of the last round become the final result of the inventor disambiguation.

A summary of the passes made over the data is provided in Table 3. On each subsequent pass, we decrease the blocking threshold; because of the record consolidation that had been applied after the previous pass, we can maintain reasonable runtimes. This allows exploration of more comparisons than would be feasible in the single-blocking scheme. Note especially the steep drop in the number of records after the first few rounds, allowing more permissive blocking.

Run #	Type	Blocking rule	Similarity Profile	Count (number of distinct inventors)
0	Preconsolidation	Exact first name. exact middle name, exact last name, city, state, country, assignee	N/A	4.51 million
1	Consolidated	First name without space, last name without space	First name, middle name, last name, city	3.09 million
2	Consolidated	First name without space, last name without space	First name, middle name, last name, coauthor, assignee, geographical location	2.84 million
3	Consolidated	First name without space, last name without space	First name, middle name, last name, coauthor, class, assignee	2.82 million
4	Consolidated	First 5 characters of first name without space, first 8 characters of last name without space	First name, middle name, last name, coauthor, geographical location, assignee	2.80 million
5	Consolidated	First 3 characters of first name without space, first 5 characters of last name without space	First name, middle name, last name, coauthor, geographical location, assignee	2.75 million
6	Consolidated	First name initial, first 5 characters of last name without space	First name, middle name, last name, coauthor, geographical location, assignee	2.70 million
7	Consolidated	First name initial, first 3 characters of last name without space	First name, middle name, last name, coauthor, geographical location, assignee	2.67 million

Table 3: Iterative blocking and consolidation scheme.

4. Results and accuracy metrics

Our goal is to properly capture and assign all of an inventor's patents to a single and unique inventor number. Analogous to type I and II error, however, no disambiguation procedure will provide perfect identification. A variety of terms have been used to measure incorrect matching, and these measures can be calculated at the record pair, patent, or inventor career level. Following Torvik and Smalheiser (2009) we use measures of lumping *L* and splitting *S,* counting the number of incorrect patent assignments. *Lumping* occurs when distinct inventors are incorrectly identified as one. *Splitting* occurs when one inventor is incorrectly identified as multiple inventors. In other words, two or more inventors in the same cluster constitutes lumping error; one inventor in two or more clusters constitutes splitting error. Lumping and splitting correlate negatively: if one goes down the other typically goes up and vice-versa.

4.1 Estimating accuracy

In order to estimate error rates, we compared our efforts to a manually curated dataset (developed from Gu et al., 2008). [17] The original dataset was a sample of 95 US inventors (1169 inventor-patent instances) drawn from the engineering and biochemistry fields, with current or previous academic affiliations. As these are eminent academics, this database oversamples prolific inventors (though this is not uncommon amongst hand curated datasets used for learning or testing purposes). The patents within the benchmark dataset were first identified from inventors' CVs. We updated these Gu et al. (2008) patent lists, and then repeatedly attempted to contact all inventors in the dataset, via email and then phone, in order to validate our disambiguation of their patents. We also cross-checked our results with online resources and human pattern recognition. We had a total of 43 confirmed responses and 52 unconfirmed responses (we differentiate between confirmed and unconfirmed in the posted file). The benchmark dataset contains the patent history of these 95 US-based academic inventors. [18]

For each inventor in this standard we identified their split records (that failed to map to his/her largest cluster). The total number of split records (44) divided by the total number of records in the standard (1169) yields our splitting statistic of 3.26%. Despite this reasonable percentage at the record level, the algorithm unfortunately splits 22 out of the 95 careers. Similarly, for each

---

[17] Jerry Marschke, lead investigator on the original development of the dataset, generously agreed to our usage and to post the results as well.

[18] The benchmark assessments are at http://funglab.berkeley.edu/benchmark_all_disamb_lump_v7_exp.php and http://funglab.berkeley.edu/benchmark_all_disamb_split_v7.php. The updated database that reflects this result is at http://funglab.berkeley.edu/pub/disamb_no_postpolishing.csv. The original database (which included a Jaro-Winkler post processing step) is at: http://dvn.iq.harvard.edu/dvn/dv/patent.

cluster in the standard, we identified lumped records (that did not belong in the largest sub-cluster by a single inventor in the standard.) The total number of lumped records (28) divided by the total number of records in the standard (1197) yields our lumping statistic of 2.34%. Only two of the 95 careers are lumped, not surprisingly, the common names of Eric Anderson and James Evans. The larger number of records in the lumped denominator reflects the inventor records outside of the standard (given an identified inventor, we collected all records with that inventor identification from the disambiguated database, thus adding 28 records). Obviously, much work remains in improving the accuracy of the disambiguation.

5. *Disambiguated data and illustrative applications*

Figure 4 shows the number of patents per unique inventor. Over half the population has only patent, and the overall distribution is skewed. Over 85% of the total inventor population has 5 or fewer patents, while less than 1% have 50 or more.
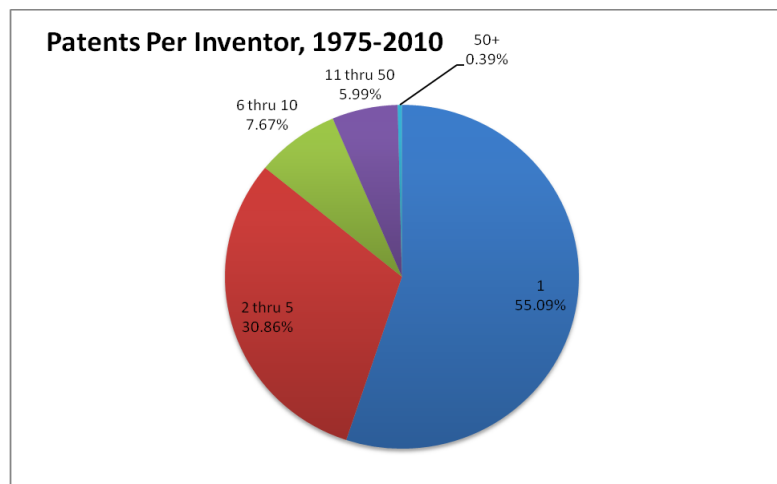
Figure 4: Number of patents per unique inventor

5.1 Inventor networks

Disambiguation of the inventor record enables research into co-authorship networks of inventors. A variety of questions can be investigated, for example, the impact of social structure on individual creativity (Fleming et al. 2007), knowledge diffusion (Singh 2005), and regional dynamics (Breschi and Lissoni 2009). Bibliometric records of co-authorship networks provide both advantages and disadvantages in the study of social structure. If the data are large enough, researchers can sample to minimize spurious significance caused by lack of independence between proximal nodes. Bibliometric networks are typically observed over time, and hence do not need to be repeatedly sampled. If the structures are large and continuous, researchers can avoid cutting networks at arbitrary points. Bibliometric networks in general are much cheaper to

build than survey networks, though they cannot capture the same richness of direct observation or survey. They avoid response bias, in that all individuals are observed, though on the other hand, they inherently suffer from selection bias, in that unsuccessful attempts to patent or publish remain unobserved.

We provide a sample of social network measures, within three-year blocks, starting in 1975 (these network measures are based on the original Harvard DVN data at http://dvn.iq.harvard.edu/dvn/dv/patent). They include degree (the number of number of unique co-authors in a three year period), eigenvector and node centrality (Bonacich 1991), and clustering coefficient (Watts and Strogatz 1998). The size of the inventor's component is also included, the number of inventors in that three-year period that can be reached through a co-author, and the ranking of this component, in the same three-year period, against all other components in that period. The Harvard DataVerse Network (DVN) interface allows researchers to subset the networks, based on a number of criteria such as name, time, or technology.

5.2 Inventor mobility movies

Much research has used patent records to study inventor mobility, often in the study of regional dynamics (Almeida and Kogut 1999, Agrawal et al. 2006; Breschi and Lissoni 2009, Marx et al. 2009). Most of this research has relied on manual or ad hoc disambiguation and not considered across-region mobility. Automated disambiguation of entire patent records enables study – and visualization – of cross-regional mobility. Figures 5-8 illustrate the emigration and immigration of the U.S. state of Michigan, in 1982, 1987, and 1987, respectively, and emigration into California, at the height of the technology boom in 2000.[19] Interestingly, early years illustrate a net loss of inventors from California, possibly due to decreased defense spending. The interested reader is encouraged to investigate all years of mobility.

Figures 5 and 6 illustrate a noticeable increase in emigration from Michigan, comparing 1982 to 1987. Figure 7 dramatically illustrates how this emigration was not balanced by immigration. Marx et al. (2012) establish that the emigration increase is partially caused by the inadvertent enforcement of noncompete covenants starting in 1985. Their identification relied on a differences-in-differences methodology, which compared emigration from Michigan to emigration from other control states that prohibited enforcement of noncompetes over the entire time period of study, from 1975-1996 (these pictures are anecdotal – we urge the interested reader to independently assess the comprehensive diffs-in-diffs models). Marx et al. also provide corroborating cross sectional evidence for all U.S. states, from 1975-2005. In total,

---

[19] Movies for all years since 1975, for these states and other states, can be viewed, along with a moving histogram of origin or destination states, at http://funglab.berkeley.edu/mobility/. The figures shown here were first published in a technical note, see Li and Fleming, 2013.

these analyses relied on the analysis of 540,780 careers; this would have been impossible without an automated and reasonably accurate disambiguation across the entire patent record.
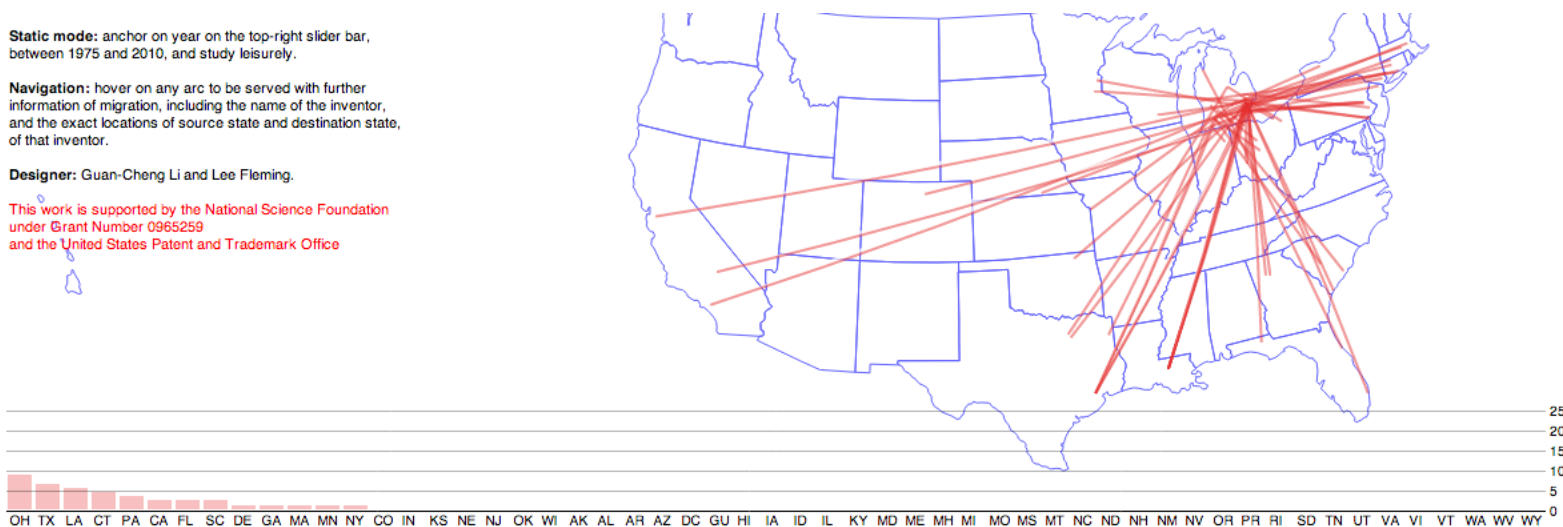
Figure 5: Emigration of patented inventors from state of Michigan in 1982.
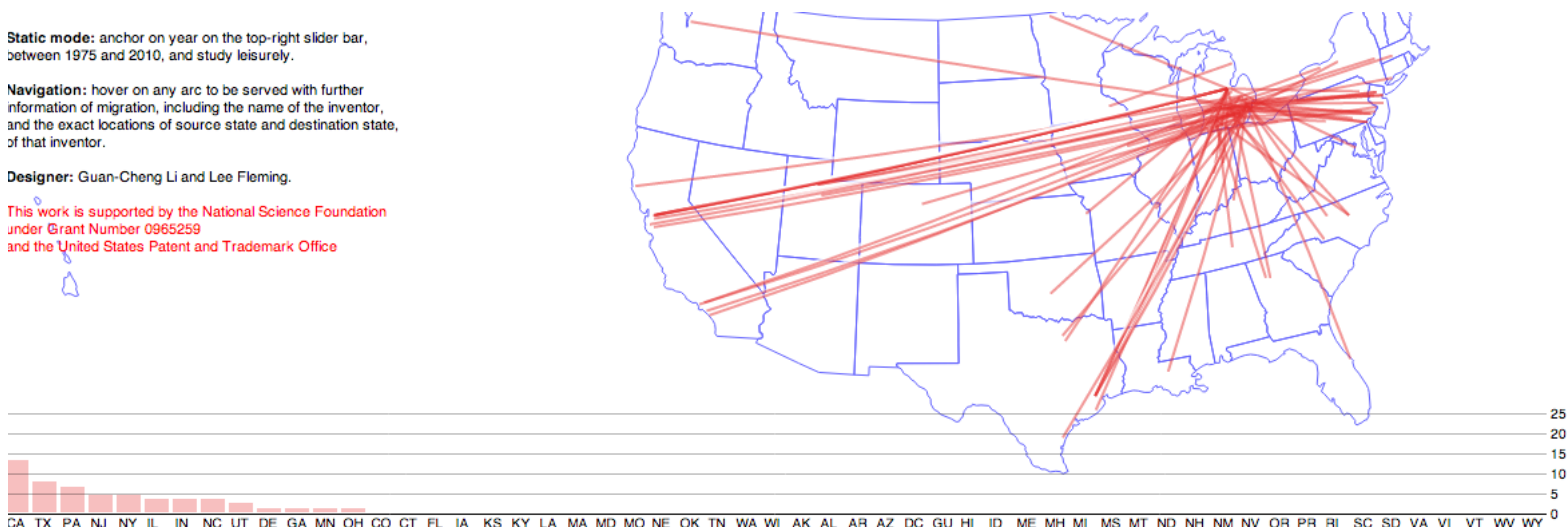

Figure 6: Emigration of patented inventors from Michigan in 1987. Note the greater total amount of emigration (the right hand tail of the distribution represents one inventor in both cases), along with the greater proportion to California, Washington, and Minnesota, states that do not enforce noncompete covenants. For comprehensive statistical evidence of a "brain-drain," please see Marx et al. 2012.

IL AK AL AR AZ CA CO CT DC DE FL GA GU HI IA ID IN KS KY LA MA MD ME MH MI MN MO MS MT NC ND NE NH NJ NM NV NY OH OK OR PA PR RI SC SD TN TX UT VA VI VT WA WI WV WY
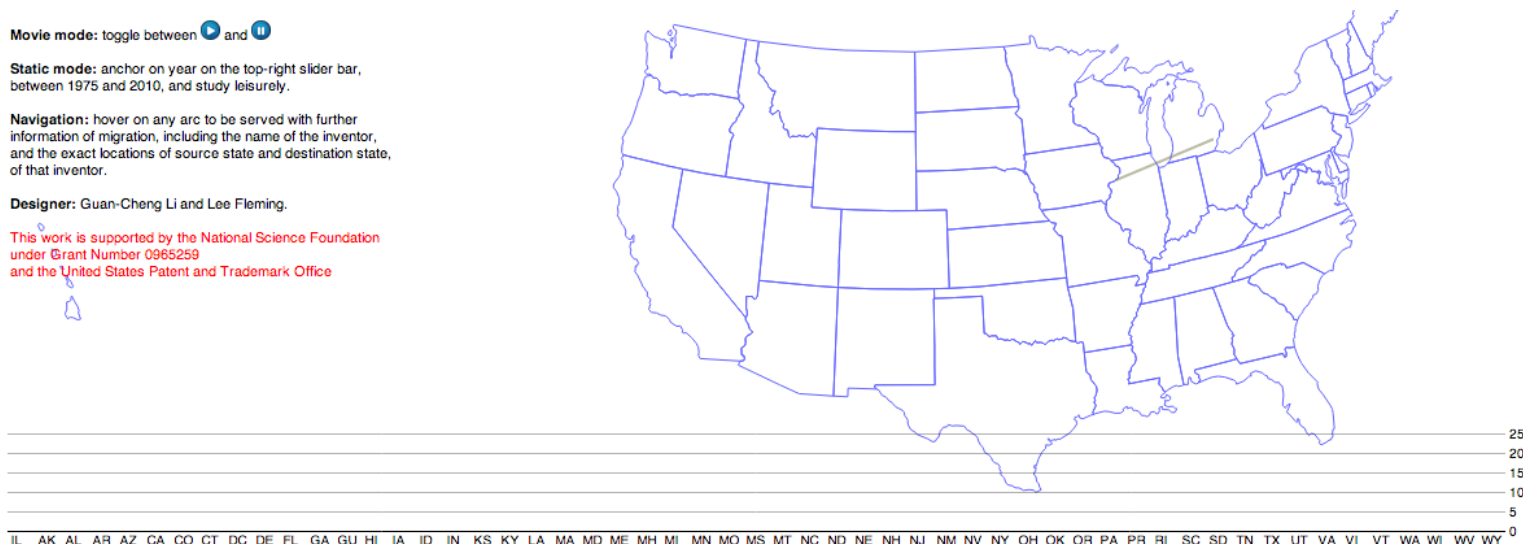
Figure 7: Immigration of patented inventors into Michigan in 1987. Note the stark contrast with emigration (Figure 6); 1987 was not an anomaly, for example, 1981 had no immigration. This reflects the general economic malaise of the state, during the contraction of the automobile industry.

NJ NY TX PA FL IL WA CO ID MA MD MN AZ CT OH MI OR MO NC WI IA GA UT VA DE IN KY LA MT NH NV RI TN MS NM OK PR VT AK AL AR CA DC GU HI KS ME MH ND NE SC SD VI WV WY

Figure 8: Influx of patented inventors into California in 2000, at height of the technology boom.

6. Conclusion

Many scholars of innovation have begun to disambiguate patent records (Fleming and Juda 2004; Singh 2005; Trajtenberg et al., 2006; Raffo and Lhuillery 2009; Carayol and Cassi, 20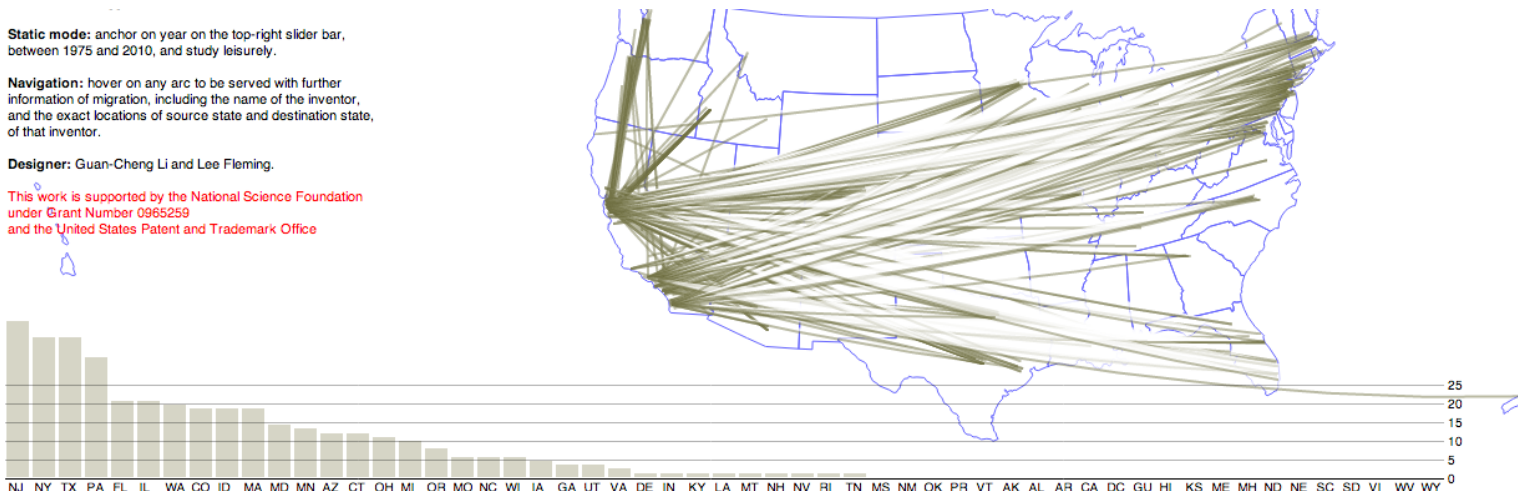09; Lai et al., 2009; Pezzoni et al., 2012). We provide a disambiguation of the U.S. patent record and make our code and algorithms public, in hopes of eliciting public use, comment, and improvement. In contrast to previous ad hoc methods, this approach drew from computer and information science (Torvik and Smalheiser, 2009) and applied a machine learning approach. The work provides public databases and tools that enable identification of co-authorship networks in the USPTO database, and an application of the data by illustrating inventor mobility into and out of Michigan, California, and other states.

6.1 Caveats and Planned Improvements

Perhaps the most important next challenge in disambiguation is to accommodate ethnic and geographical differences; we have adopted a U.S. centric approach, and not surprisingly, European names consequently work best. Chinese, Korean and Taiwanese generally do not have middle names although a western style "middle name" can still be extracted. Their first names can be reset to the concatenation of extracted first name and extracted last name; their last name the same; their middle names to the concatenation of modified first name and last name. For Japanese names, however, the raw data generally do not contain a middle name and the names are usually very similar in their English spellings. Related to these challenges, some first and last names, even non-Asian names, can be switched in the input data.

While our work is based mostly on the 2005 Author-ity model (Torvik et al. 2005), more recent work in 2009 (Torvik and Smalheiser 2009) provides a number of suggestions for more rigorously setting parameters like block priors and the weighting coefficient in triplet correction, and for handling correlations between fields in the data (e.g. living in Korea and working for Samsung) that can bias disambiguation results. Other potential improvements include: accounting for firm size in the assignee comparison algorithm, incorporating population density as an additional factor for the location comparison algorithm, and using additional data fields (essentially expanding the profile feature set), such as comparisons of titles and abstracts or patent lawyers and prior art citations (Tang and Walsh, 2010). Also, existing data fields such as technology sub-classes and co-authors could be examined in finer detail. Ideally, scholars might choose from amongst multiple disambiguated datasets, each of which would avoid using the variable of interest for disambiguation (for example, if a researcher was studying inventor mobility across firms, the database would ideally not use assignees in disambiguation, for an example of this approach based on simulation, see Pezzoni et al. 2012). Much work remains; hopefully this disambiguation and the public data it creates can provide the foundation for future improvements and increased research productivity.

References

Agrawal, A., I. Cockburn, and J. McHale, 2006. Gone but not forgotten: knowledge flows, labor mobility, and enduring social relationships. *Journal of Economic Geography* 6(5).

Almeida, P. and B. Kogut, 1999. Localization of Knowledge and the Mobility of Engineers in Regional Networks. *Management Science* 45(7):905-917.

Bishop, C. 2006. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, NY.

Bilenko, M., Kamath, B., and Moody, R. J., 2006. Adaptive Blocking: Learning to Scale Up Record Linkage, *Proceedings of the Sixth IEEE International Conference on Data Mining (ICDM-06)*, 87-96.

Bonacich, P., 1991. Simultaneous group and individual centralities. *Social Networks* 13, 155–168.

Breschi, S. and Francesco L., 2009. Mobility of skilled workers and co-invention networks: an anatomy of localized knowledge flows. *Journal of Economic Geography* 9(4):439-468.

Carayol N., and Cassi L., 2009. Who's Who in Patents. A Bayesian approach. *Cahiers du GREThA 2009-07*.

Diesner, J. and K. Carley, 2009. He says, she says. Pat says, Tricia says. How much reference resolution matters for entity extraction, relation extraction, and social network analysis. *IEEE Symposium: Computational Intelligence for Security and Defence Applications*, Ottawa, Ca.

Fegley BD, Torvik VI (2013) Has Large-Scale Named-Entity Network Analysis Been Resting on a Flawed Assumption? PLoS ONE 8(7): e70299.

Fleming, L. and A. Juda, 2004. A Network of Invention. *Harvard Business Review* 82: 6.

Fleming, L., 1991. VLSI chip having improved test access. US patent 5,029,133.

Fleming, L., 1992. Local tristate control circuit. US Patent 5,136,185.

Fleming, L. and S. Mingo, D. Chen, 2007. Collaborative Brokerage, Generative Creativity, and Creative Success. *Administration Science Quarterly,* 52: 443-475.

Gu, Gyojeong; Lee, Sangjoon; Kim, Jinyoung; and Marschke, Gerald, 2008. Matching Accuracy of the Lee-Kim-Marschke Computer Matching Program. *SUNY Albany working paper*, March 2008.

Hall, B. H., A. B. Jaffe, and M. Trajtenberg, 2001. The NBER patent Citations Data File: Lessons Insights and Methodological Tools, NBER.

Hastie, T., Tibshirani, R., and Friedman, J. H. 2001. *Elements of Statistical Learning: Data Mining, Inference and Prediction (Second Edition).* Springer-Verlag, New York, NY.

Hazewinkel, M., ed. (2001), "Laplace transform", Encyclopedia of Mathematics, Springer, NY, NY.

Herzog, T., F. Scheuren and W. Winkler, 2007. *Data Quality and Record Linkage Techniques.* New York, Springer Press.

Johnson, K. 2013. "Inferring location data from United States Patents," Fung Institute Technical note (forthcoming).

Lai, R., A. D'Amour and L. Fleming, 2009. 2009-02-18, "The careers and co-authorship networks of U.S. patent-holders, since 1975", http://hdl.handle.net/1902.1/12367 UNF:5:daJuoNgCZlcYY8RqU+/j2Q== Harvard Business School; Harvard Institute for Quantitative Social Science.

Lewis, D. D., 1998. Naïve (Bayes) at Forty: The Independence Assumption in Information Retrieval. In *ECML '98 Proceedings of the 10th European Conference on Machine Learning.* pp. 4-15, Springer-Verlag, London.

Li, G. and L. Fleming. 2013. MobilityMapper: Visualizations of U.S. Inventor Mobility, 1975 – 2010. Fung Institute Technical Note.

Marx, M. and D. Strumsky and L. Fleming, 2009. Mobility, Skills, and the Michigan Non-compete Experiment. *Management Science*, 55: 875-889.

Marx, M., J. Singh and L. Fleming, 2012. Regional Disadvantage? Non-competes and Brain-Drain. *Working paper, MIT Sloan School and Harvard Business School.*

On, B.-W., Lee, D., Kang, J., and Mitra, P., 2005. Comparative Study of Name Disambiguation Problem Using a Scalable Blocking-based Framework. *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, 334-353.

Pezzoni, M. and F. Lissoni, G. Tarasconi, 2012. "How To Kill Inventors: Testing The Massacrator© Algorithm For Inventor Disambiguation." *Cahiers du GREThA* n° 2012-29. http://ideas.repec.org/p/grt/wpegrt/2012-29.html.

Raffo, J. and S. Lhuillery, 2009. How to play the "Names Game": Patent retrieval comparing different heuristics. *Research Policy* 38, 1617–1627.

Reuter, T., and Cimiano, P., 2012. A Systematic Investigation of Blocking Strategies for Real-Time Classification of Social Media Content into Events. *Association for the Advancement of Artificial Intelligence Technical Report WS-12-02*, p. 8-15.

Rish, I., 2001. An Empirical Study of the Naïve Bayes Classifier. *IBM Research Report* RC 22230 (W0111-014).

Singh, J., 2005. Collaborative networks as determinants of knowledge diffusion patterns. *Management Science*, 51(5): 756–770.

Singh, J. and L. Fleming, 2010. "Lone Inventors as Sources of Technological Breakthroughs: Myth or Reality?" *Management Science*, 56(1): 41-56.

Smalheiser, N. R., Torvik, V. I., 2009. Author name disambiguation. In *Annual Review of Information Science and Technology* (B. Cronin, Ed.), Vol. 43, pp. 287-313.

Tang, Li; Walsh, John P., 2010. Bibliometric fingerprints: name disambiguation based on approximate structure equivalence of cognitive maps. *Scientometrics* **84**:763-784.

Torvik, V. and M. Weeber, D. Swanson, N. Smalheiser, 2005. A Probabilistic Similarity Metric for Medline Records: A Model for Author Name Disambiguation. *Journal of the American Society for Information Science and Technology,* 56(2):140–158.

Torvik, V. and N. Smalheiser, 2009. Author Name Disambiguation in MEDLINE. *ACM Transactions on Knowledge Discovery from Data*, Vol. 3., No. 3, Article 11.

Trajtenberg, M., G. Shiff, and R. Melamed, 2006. The Names Game: Harnessing Inventors Patent Data for Economic Research. NBER.

Watts, D. and S. Strogatz 1998. "Collective dynamics of 'small-world' networks". *Nature* 393 (6684): 440–442.

Zhang, H., 2004. The Optimality of Naïve Bayes. In *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference*, Miami Beach, Florida, USA.

Further USPTO XML file clarification

This Appendix is based upon correspondences with the USPTO and further clarifies the XML patent file. The language used here was provided by the USPTO.

Appendix 1 Table 1 - U.S. Patent Grant and Published Applications Document Numbers:

Patent Grant Patent Number

- Design Patents
  - Position 1 – A constant "D" identifying the granted document as a Design Patent.
  - Positions 2-8 – Seven-position numeric, right justified, with a leading zero.
- SIR Patents
  - Position 1 – A constant "H" identifying the granted document as a Statutory Invention Registration (SIR).
  - Positions 2-8 – Seven-position numeric, right justified, with a leading zero.
- Plant Patents
  - Positions 1-2 – A constant "PP" identifying the granted document as a Plant Patent.
  - Positions 3-8 – Six-position numeric, right justified, with a leading zero.
- Reissue Patents
  - Position 1-2 – A constant "RE" identifying the granted document as a Reissue Patent.
  - Positions 3-8 – Six-position numeric, right justified, with a leading zero.
- Utility Patents
  - Positions 1-8 – Eight-position numeric, right justified, with a leading zero.
- X-Series
  - Patents issued between July 31, 1790 and July 4, 1836. They were not originally numbered, but have since been assigned numbers in the sequence in which they were issued
  - Positions 1-8 – Eight-position, right justified, with a leading "X".

Appendix 1 Table 2 - U.S. Patent Grants and Patent Published Applications

Kind Codes

Note: The following 2-position kind codes will be present in the XML <kind> tags of Red Book and Yellow Book. These 2-positions kind codes will also be present on the printed documents

with the following exceptions: Reissues will contain a single position "**E**", SIR documents will contain a single position "**H**", and Designs will contain a single position "**S**".

- A1 - Utility Patent Grant issued prior to January 2, 2001.
- A1 - Utility Patent Application published on or after January 2, 2001
- A2 - Second or subsequent publication of a Utility Patent Application
- A9 - Correction published Utility Patent Application
- Bn - Reexamination Certificate issued prior to January 2, 2001.
    - NOTE: "n" represents a value 1 through 9.
- B1 - Utility Patent Grant (no published application) issued on or after January 2, 2001.
- B2 - Utility Patent Grant (with a published application) issued on or after January 2, 2001
- Cn - Reexamination Certificate issued on or after   January 2, 2001.
    - NOTE: "n" Represents a value 1 through 9 denoting the publication level.
- E1 - Reissue Patent
- H1 - Statutory Invention Registration (SIR) Patent Documents.
    - Note: SIR documents began with the December 3, 1985 issue
- I1 - "X" Patents issued from July 31, 1790 to July 13, 1836
- I2 - "X" Reissue Patents issued from July 31, 1790 to July 4, 1836
- I3 - Additional Improvements – Patents issued issued between 1838 and 1861.
- I4 - Defensive Publication – Documents issued from Nov 5, 1968 through May 5, 1987
- I5 - Trial Voluntary Protest Program (TVPP) Patent Documents
- NP - Non-Patent Literature
- P1 - Plant Patent Grant issued prior to January 2, 2001
- P1 - Plant Patent Application published on or after January 2, 2001
- P2 - Plant Patent Grant (no published application) issued on or after January 2, 2001
- P3 - Plant Patent Grant (with a published application) issued on or after January 2, 2001
- P4 - Second or subsequent publication of a Plant Patent Application
- P9 - Correction publication of a Plant Patent Application
- S1 - Design Patent

Appendix 1 Table 3 - U.S. Application Series Codes

Code:	Filing Dates:
02	Filed prior to January 1, 1948
03	January 1, 1948 through December 31, 1959
04	January 1, 1960 through December 31, 1969
05	January 1, 1970 through December 31, 1978
06	January 1, 1979 through December 31, 1986
07	January 1, 1987 through January 21, 1993
08	January 22, 1993 through January 20, 1998

09	January 21, 1998 through October 23, 2001
10	October 24, 2001 through November 30, 2004
11	December 1, 2004 through December 5, 2007
12	December 6, 2007 through Current

Design Patents

Code:	Filing Dates:
07	Filed prior to October 1, 1992
29	Filed after October 1, 1992

Note: The Design Series Coded "29" is present in the XML data as "29" and is displayed as a "D" on Patent on the Web.

Appendix 1 Table 4 - U.S. Patent Classifications

Class

- A 3-position alphanumeric field right justified with leading spaces.
- Design Patents
    - The first position will contain a "D".
    - Positions 2 and 3, right justified, with a leading space when required for a single digit class.
- Plant Patents
    - Positions 1-3 will contain a "PLT"
- All Other Patents
    - Three alphanumeric positions, right justified, with leading spaces

Sub-Class

- Three alphanumeric positions, right justified with leading spaces, and, if present, one to three positions to the right of the decimal point (assumed decimal in the Red Book XML), left justified.
- A digest entry as a sub-class would appear as follows:
    - Three positions containing "DIG", followed by one to three alphanumeric positions, left justified.

Appendix 1 Table 5: Assignee Type Categories

01 Unassigned
02 United States company or corporation

03 Foreign company or corporation

04 United States individual

05 Foreign individual

06 U.S. Federal government

07 Foreign government

08 U.S. county government

09 U.S. state government

Categories 10-16 are currently unexplained by the USPTO. Source:
http://www.uspto.gov/web/offices/ac/ido/oeip/sgml/st32/redbook/pap-v15-2001-01-31/dtdelem/assignee-type.html.

Appendix 2 Data distribution

All the data used in and resulting from the disambiguation is public and freely available through the Harvard Dataverse Network. Supporting datasets contribute either to creating the consolidated inventor results dataset or enhance the algorithm. Other datasets derived from parsing USPTO patent data are included for reference. Due to the portability of the file type, we now employ Sqlite3 for database files. Results datasets are presented in both Sqlite3 and .csv formats.

The Harvard Patent Dataverse provides a platform for providing access to the various datasets described in this paper. Files are also available on the Fung Institute GitHub website.

1. Raw Patent Datasets consisting of individual zipped directories containing parsed USPTO patent data in sqlite3 and .csv formats.
2. Network datasets consisting of individual subsettable GraphML files for every three years from 1975-2010. Networks consist of inventors as nodes and patents as links.
3. Results datasets consisting of individual subsettable tabular datasets for every three years from 1975-2010. Includes inventor and patent data, and calculated variables.
4. Full disambiguation results including individual zipped directory containing sqlite3 and .csv files.
5. Benchmark dataset used for results analysis.

Appendix 3 Software and computation

We wrote a generic disambiguation engine in C/C++, in order to provide developers with a modular and computationally efficient way to specify any disambiguation strategy on any database. Quadratic programming for the interpolation, the extrapolation and the enforcement of monotonicity of similarity profiles is performed using IBM CPLEX. It takes about three hours to concurrently complete the adjustment of the six dimensional similarity profiles on an 8CPU 24GB workstation. The original code base is currently available online at http://www.GitHub.com/patentnetwork/CPP_Disambiguation. Revised and updated code is available at https://GitHub.com/funginstitute/downloads. We invite community members to use this implementation to write their own disambiguation of the patent database.