FUNG INSTITUTE
for ENGINEERING LEADERSHIP
UC BERKELEY COLLEGE OF ENGINEERING

Disambiguation of Inventors, USPTO 1975 – 2013

Guan-Cheng Li
University of California, Berkeley

College of Engineering
University of California, Berkeley

Fung Technical Report No. 2013.09.17
http://www.funginstitute.berkeley.edu/sites/default/files/USPTO.pdf

September 17, 2013

The Coleman Fung Institute for Engineering Leadership, launched in January 2010, prepares engineers and scientists – from students to seasoned professionals – with the multidisciplinary skills to lead enterprises of all scales, in industry, government and the nonprofit sector.

Headquartered in UC Berkeley's College of Engineering and built on the foundation laid by the College's Center for Entrepreneurship & Technology, the Fung Institute combines leadership coursework in technology innovation and management with intensive study in an area of industry specialization. This integrated knowledge cultivates leaders who can make insightful decisions with the confidence that comes from a synthesized understanding of technological, marketplace and operational implications.

Abstract: Since the USPTO does not require unique identifiers in the application process, it can be hard to search for a specific inventor and his or her patents, especially if the inventor's name is common or has multiple forms. Ad-hoc disambiguation methods based on thresholds and string comparison matching are common but vulnerable to bias and idiosyncratic to a training set. To tackle the problem, I employ a large scale-clustering algorithm, adaptive K-means, which automates the disambiguation. The algorithm considers statistical correlations between each inventor name pair, and learns from the entire USPTO dataset regarding when that pair needs to be lumped, i.e., be treated as the same person, thereby being assigned one unique identifier, or to be split, i.e., be treated as different persons, hence being assigned two different identifiers. Our goal is to have a robust engine that will automatically disambiguate the entire USPTO inventors on a weekly basis, upon each issuance of new patents every Tuesday, and to make the disambiguated dataset available to interdisciplinary researchers who rely on this data.

This work is supported by the United States Patent and Trademark Office
And the American Institutes for Research.

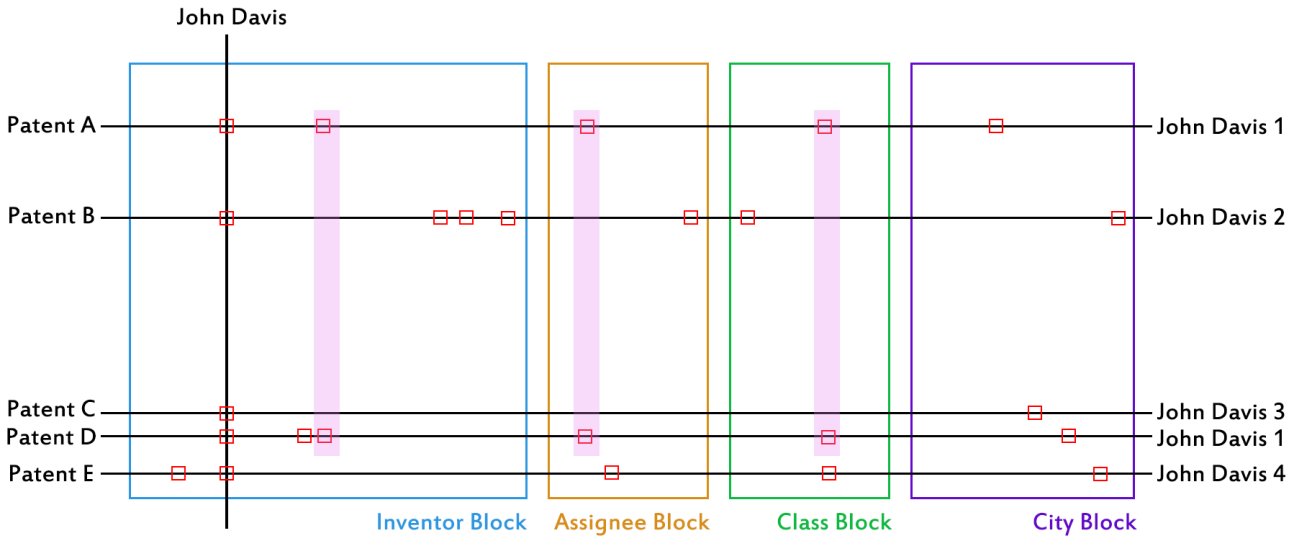
130 Blum Hall #5580 Berkeley, CA 94720-5580 | (510) 664-4337 | www.funginstitute.berkeley.edu

Algorithm

To disambiguate inventors, we consider several patent attributes including the published name, patent technology class, city, his/her co-inventor names, and assignee. To disambiguate, we mean to assign unique identifiers to each inventor across the USPTO patent database.

Step 1: Vectorization

We define a document unit of a patent as a collection of that patent's attributes. We represent that unit as an unordered collection of keys. The occurrence of each key is used as a feature for training a classifier. We build a document-by-key incidence matrix, which is sparse because a patent cannot use all the keys. For example, if a patent has three inventors, has a primary class, has one assignee, has a city, then the sum of the row of that patent is 3+1+1+1=6. If a patent is filed by a lone inventor, then the sum of the row of that patent is 1+1+1+0=3. To illustrate, see the below figure.



Suppose John Davis filed five patents, A, B, C, D, E. Here, let John Davis be a column that is depicted by a black vertical tab. We look closer at the five rows that have '1's along that column, namely the index of the patents being filed by John Davis.

Having formed a matrix, we can compute correlations between rows (patents) and columns (inventors).

Step 2: Distance measurements

Distance measurements can be computed in a number of ways, e.g., Euclidean distance, Manhattan distance, Chebyshev distance, and Mahalanobis. They allow for evaluation of the

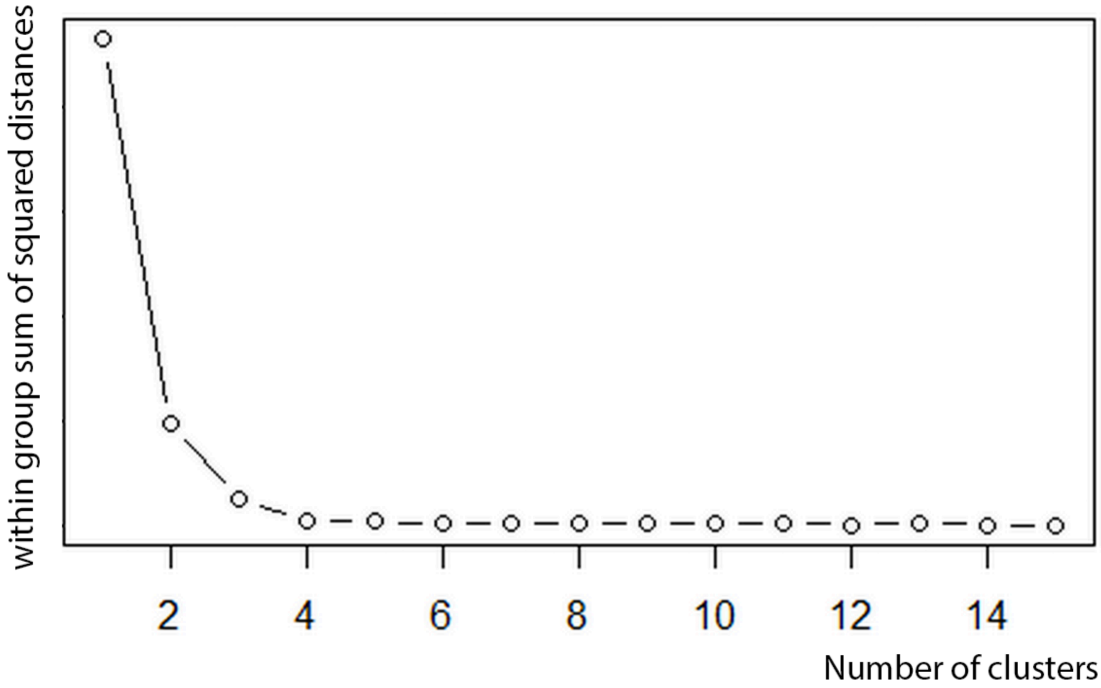
distance that an inventor varies from the other inventor. Here, I adopt the Euclidean distance.

$$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T \Sigma^{-1} (\vec{x} - \vec{y})}.$$

Step 3: Splitting the same inventor name by inter-clustering

Certain common names can represent multiple persons, e.g., John Davis. If, in fact, there are four different such people across the USPTO dataset, we should prepare four unique identifiers for John Davis.

We examine the inventor block and extract a list of unique inventor names. To initialize, we treat each individual inventor as a distinct person, e.g., by assigning initial identifiers. Then, we cluster each inventor block that is centered by the inventor names, i.e., John Davis as an example, by applying the k-means clustering algorithm based on a bottom-up approach. When splitting, we calculate the new means to be the centroids of the observations in the new clusters.

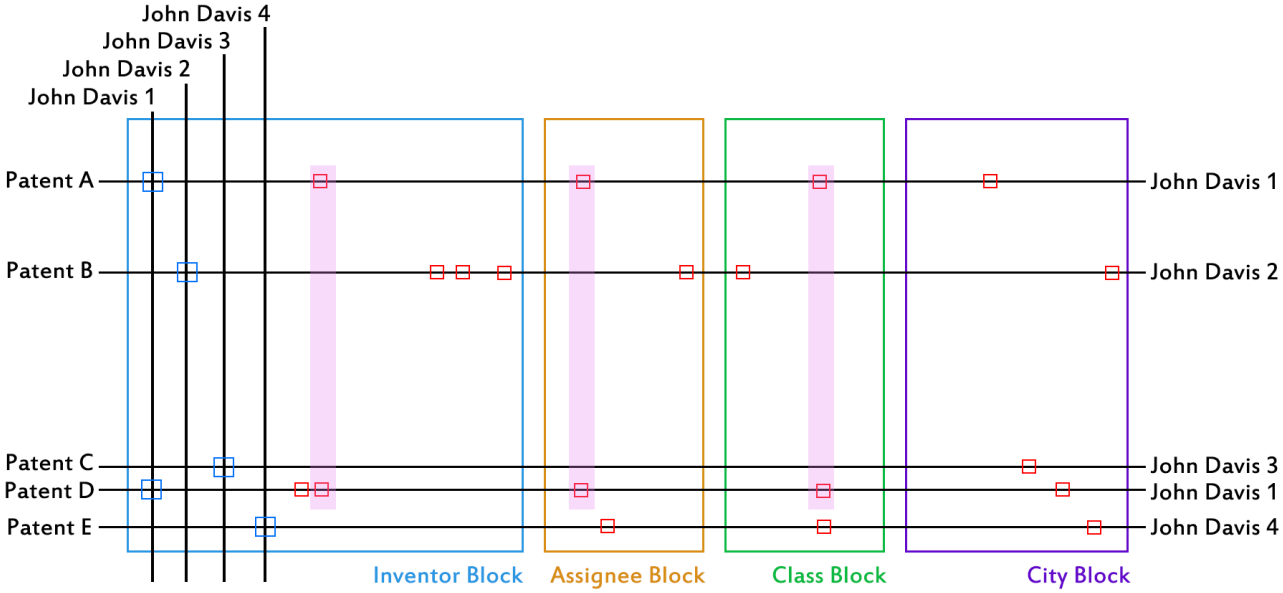


We determine the cluster size by minimizing the sum of each clusters' sum of the inter-cluster's squared distances. In this example, the five names, John Davis, are grouped down to four clusters. When the sum no longer decreases, we have reached a local minimum. In other words, increasing the cluster number to five wouldn't decrease the objective function and hence we stop at four clusters for John Davis.

Step 3: Lumping different inventor names by intra-clustering

Once splitting is done, we assign the centers of each cluster unique identifiers and augment the

matrix column-wise by their unique identifiers, as depicted



By lumping, we mean to merge naming aliases into one person if, in fact, they should be one person based on what the other factors, such as co-inventor, assignee, class, or city, suggest. This step is designed to treat Jim and James, Bob and Robert, or, Arnaud Gourdol, Arnaud P Gourdol, Arnaud P J Gourdol, and Arno Gourdol as same persons. The algorithm assigns inventors to the nearest cluster by distance. Here, if an inventor not in a cluster is determined by the algorithm to be lumped into another cluster, there are three possibilities for the naming match.

(1) The last names agree, the first names disagree, and the first letter of the middle names agree (if any), for example, Jim and James. Lumping is performed.

(2) The last names disagree, the first names agree, and the first letter of the middle names agree (if any), due to marriage and last name change. Lumping is performed.

(3) Both the last names and first names disagree. Name change is the only reason for them to be the same person. Without personally knowing one of the inventors, it is impossible to tell. Lumping is not performed.

Step 4: Blocking and tie-breaking

The goal of the automated K-mean algorithm is to produce a high cluster quality with high intra-cluster similarity and low inter-cluster similarity. This objective function to be optimized is expressed as:

$$\phi^{(Q)}(\mathbf{X},\lambda) = 1 - \frac{\sum_{i=1}^{k} \frac{n_i}{n-n_i} \sum_{j\in\{1,\ldots,i-1,i+1,\ldots,k\}} n_j \cdot \mathrm{inter}(\mathbf{X},\lambda,i,j)}{\sum_{i=1}^{k} n_i \cdot \mathrm{intra}(\mathbf{X},\lambda,i)}$$

$Ø^Q$ represents the cluster quality. If the quality is 0 or lower, then two items of the same cluster are, on average, more dissimilar than a pair of items from two different clusters. If the quality rating is closer to 1, it means that two items from different clusters are entirely dissimilar, and items from the same cluster are more similar to each other. This will also result in a denser k-mean.

Generalization

The vectorization of entire USPTO database allows for the disambiguation process to be generic. This will also facilitate disambiguation of attorney names by examiner names by simply augmenting the matrix to the right by attorney blocks or examiner blocks, or tags.

Results for Download

The disambiguation engine processed a total of 10,708,200 inventor names across 5,021,243 patents, and identified 3,421,276 unique inventors. Download

1975 – July 16, 2013

http://funglab.berkeley.edu/pub/uspto_inventor_disambiguated_201307016_x.csv

http://funglab.berkeley.edu/pub/uspto_inventor_disambiguated_201307016_x.sqlite3

Accuracy Assessment

We assess accuracy by measuring lumping *L* and splitting *S* errors. Lumping occurs when distinct inventors are incorrectly identified as one. Splitting occurs when one inventor is identified as multiple inventors. In the present method, two or more inventors in the same cluster constitutes a lumping error; one inventor in two or more clusters constitutes a splitting error.

In order to estimate the error rates in the two clustering solutions, we compared our efforts to a manually curated dataset. The original dataset was a sample of 95 US inventors (1333 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. The patents within the benchmark dataset were first identified from inventors' CVs. We 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. The latest version of the comparison file is downloadable at:

http://funglab.berkeley.edu/guanchengli/GoldenListV7.xlsx
(Benchmark file V7)

The actual comparison results are at:
http://funglab.berkeley.edu/guanchengli/benchmark_v7_new_engine.txt

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 divided by the total number of records in the standard yields our splitting statistic. Similarly, for each 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 divided by the total number of records in the standard yields our lumping statistic. Based on this benchmark, splitting and lumping errors are 11/558 = 1.97% and 27/585 = 4.62%.