Taxonomy of US patents

'A Needle in a Data Haystack' (67978) course

- Final Project -

Motti Gold* Maria Dyshel† Sria Louis‡

February 27, 2017

Abstract

The USPTO patent archive consists of millions of patents. In addition to the patent article, each patent is citing previous related patents. The different patents were historically categorized manually in a complex hierarchical categorization system. In this paper we describe an approach to create a categorization based on textual features of the patent (i.e. title, abstract etc.) and features extracted from the citation graph.

1 The Data

The data consists of 5M patent records, registered at the United States Patent and Trademark Office (USPTO) between February 1976 and July 2014, about 3,000 patents per week.

Each patent consists of :

1. Title
2. Abstract : Detailed (technical) description of the patent
3. Citations : List of patents cited by the current patent
4. Authors names
5. Classification

The raw data, as csv files is 5.12GB.

2 Problem Description

An important aspect of registering a patent is its categorization which can be used for patents search and comparison. As of today, patents are manually categorized both by the patentees and by USPTO. clearly, patentees and regulatory entities have different - and possibly conflicting - interests and are categorizing the similar patents in far and disagreeing classes.

In addition, United States Patent Classification (USPC), which was used for patent classification until December 2012, is an over-complicated and unintuitive system, with large number of redundant classes. As a result, today categorization is archaic and hard to use; Manual re-categorization of the patents is a labor-intensive and time-consuming task. In this light, automatic categorization of the patent can present a particular interest.

In this paper we are suggesting a method to automatically re-categorize the US patents and, more generally, we want to automatically create a taxonomy of all patents.

*mottig@cs | e-mail: mottig@mail.huji.ac.il .

†smdyshel@cs | e-mail: maria.dyshel@cs.huji.ac.il .

‡slouis@cs | e-mail: sria.louis@cs.huji.ac.il .

3 Our Solution

3.1 In a nutshell

The uniqueness of the current problem is that the data consists of both text, e.g. the abstracts, and a network (the citations graph). Can we make categorization based on these two perspectives? Leveraging these two aspects as different features, we made the textual and the graph features of hierarchical clustering methods to result a category dendrogram.

3.2 Notations and Terminology

The patents: Out of more than 5M patents, we cleaned from our dataset patents lacking information (e.g. patents without a title, or patents that were only cited) and that left 3,375,627 patents. We will denote: n =3,375,627 , the number of patents. Some of the patents removed are important, although we don't know much about them, they can be cited by patents in our dataset. Therefore, the citation graph contains much more patents than n:

The Citation Graph : Patents are citing other patents. We can define citation graph $G =< V, E >$, where the vertices V are the patents and the edges are citations: $(p_1, p_2) \in E \iff p_1$ *is citing* p_2 In our dataset: $|V| = 5,759,230$ and $|E| = 57,584,272$ Note that indeed $|V| > n$.

Base Clusters: The leaves of the categories dendrogram, i.e., the sub-categories. The number of base clusters, denoted r , is the "maximal resolution" of our categorization.

Dendrogram: The tree-like object with the following layers: 1) root (all patents), 2) the last layer (i.e., leaves) which represent the base clusters (the sub-categories), 3) all other layers between the previous tow, contians the hierarchical clustres of the patents.

$\boldsymbol{n}$: Number of patents to be categorized.
G: The citation graph.
V: The set of patents in the citation graph
E: The set of the citations
r: Number of base-clusters

3.3 Zoom Into the Data

In this paper, we suggest a way to automatically create taxonomy for patents based on the two main approaches:

- ***Textual Clustering***: For instance, patent A, written in 1990 for a touch-screen technology, and patent B, written in 2016, for medical device with innovative usage of a touch-screen, might be categorized far - but will use similar nouns ("screen" / "pixle" / "finger"). Hence, even though USPTO categorize the patent very far, we can find textual similarities.
- ***Citation Graph Community Detection***: In the above example, both patent A and patent B, will cite previous patents, in the field of touch-screen technologies. We can run community detection algorithms on the citation graph to find that patents A and B have many common patents cited or that they are cited by many patents from the same field.

3.4 The Scheme of the Algorithm

1. Feature Extraction : We created feature space based on both textual and graph features
2. K-means algorithm to get base clusters
3. On these base clusters we run hierarchical clustering to get a dendrogram.

3.5 Algorithm Pipeline

1. ***Feature Extraction***: We created four feature matrices, 2 textual and 2 graphical:
    - Two textual feature matrices: We ran *tf-idf* over the patent **abstracts** and the patent **titles** (after removing stop-words etc.), that results in 2 large tf-idf matrices.

- Two graphical feature matrices: We used two algorithms for community detection over large sparse graphs. Clearly, communities with 1 patents are less informative and consider here as redundant. The non-redundant communities were translated into binary dummy variables ("One-hot" matrices), resulting a very sparse matrices with dimension (n, N_c) where N_c is the number of non-redundant communities. The two community detection algorithms we used are:
  (a) *python-igraph community label propagation* [1] (runs in $O(|E| + |V|)$)
  (b) *python-igraph community multilevel* [2] (runs in (ad-hoc linear on 'typical' real networks $O(|E| + |V|)$)

2. Feature Space Dimension Reduction:
   - Concatenate the textual feature matrices resulting textual feature matrix T
   - Concatenate the graph feature matrices resulting textual feature matrix M
   - *sklearn TruncatedSVD* and normalize each of the features matrix T and M [3]
   - concatenate $A = MT$: resulting tall matrix of dimensions (n, f) where f is the dimension of (reduced) features space.

3. Clustering:
   - run *sklearn.MiniBatchKMeans* over the whole data set A to get r base clusters: resulting small matrix S of dimensions (r, f).
   - run *scipy.cluster.hierarchy.linkage* on S to get hierarchal clustering of the r clusters: resulting z-matrix of the hierarchal clustering.
   - run *scipy.cluster.hierarchy.dendrogram* on the z-matrix to visualize the hierarchal clustering.

3.6 Packages and Parameters

3.6.1 Choosing packages

Both *sklearn TruncatedSVD* and *sklearn.MiniBatchKMeans* were chosen as more conservative algorithms (PCA and K-means) failed on our data due to memory limit or time limits. Deterministic community detection algorithms clearly failed on 5M vertices and 50M edges.

3.6.2 Choosing parameters

As part of the algorithm, we chose many parameters that we found working well. Fine-tunning of those parameters is interesting question and might improve the results in different configurations (or can be taken as different features).

- tf-idf : we used 1-grams and in addition to the basic stop-words, we removed words that typical to patents (such as "Method" and "apparatus").
- Caveat: Running tf-idf on both *abstract* and *title* as two different features was found to lose accuracy, probably because the title is less informative and contained in the abstract. Using the title with caution can improve the result with.
- Community Detection: We took undirected unweighted graph. The vertices (patents) and the edges (citations) can be weighted according to patent times, authors and other external data.
- The number of layers in the low dendrogram was chosen to be 8 in first layer and $r = 639$ in the base-layer was chosen to be equal to the USPTO new classification system (IPC).

4 Results

To measure the accuracy of the classification algorithm, 100 pairs of articles were manually labelled by the project members by answering the question: "Can the two patents belong to the same major section in patent classification?". As a guideline, a section list of IPC system was used. After applying the classification algorithms, an ROC was created by varying the distance threshold that is used by fcluster() function to create flat cluster structure from the hierarchical clustering results. Fig. 1 presents the resulting ROC curves for using graph feature set, text feature set, and both feature sets, to create the classification; as can be seen from the graph, the graph features perform notably better that the text features, and combining the two features together does not significantly improve the classification accuracy; the AUC for the three algorithms are 0.57 for text features, 0.69 for graph features and 0.70 for combined feature set.

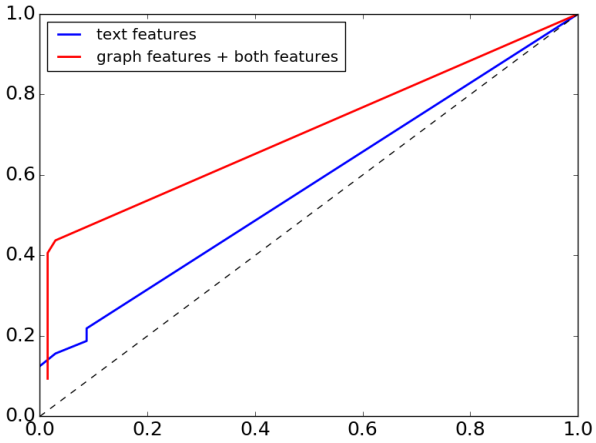

Figure 1: The ROC curve of classification prediction

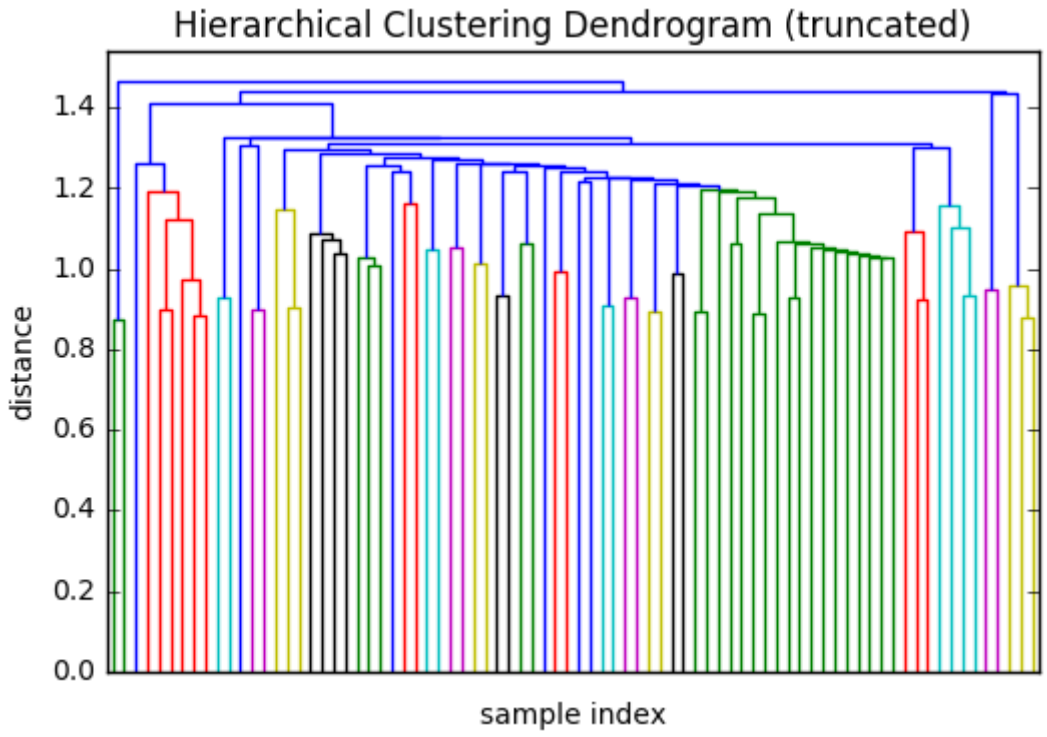

Figure 2: zoom over the top layers of the dendrogram

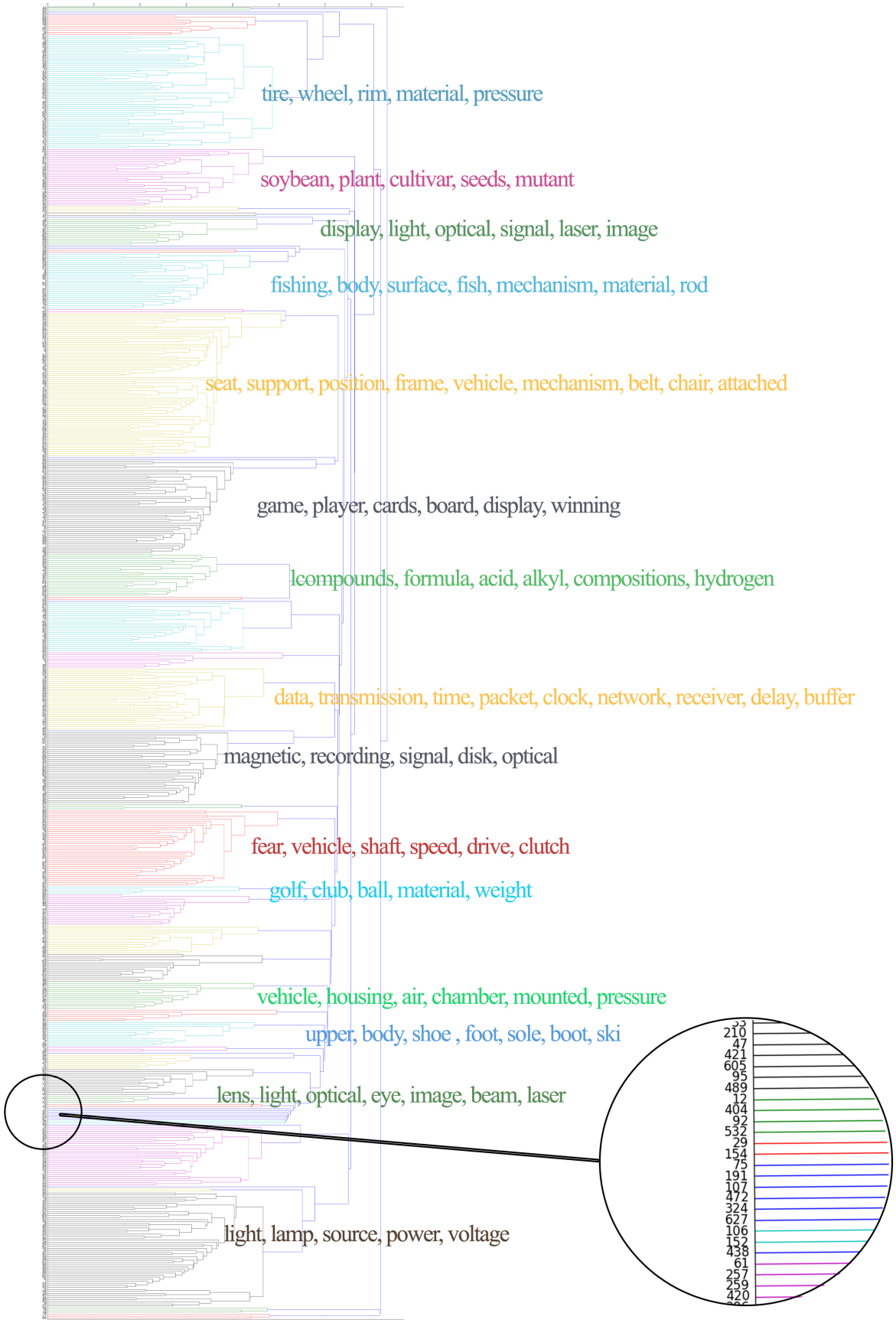

Figure 3: Full dendrogram of the automatic classification, with 639 categories in the base-clusters. The colours are helpful, to to recognize close clusters. Some of the most common words in a close clusters are shown for illustration purpose.

5 Evaluation

The evaluation of the project's output was found to be one of the trickiest parts of this project, for a number of reasons. First, it should be clear that there is no one correct solution for classification challenge; multiple hierarchies could be imposed on the patent set. This is further complicated by the fact that the hierarchy of the patents is not necessarily tree-like, but rather directed-acyclic-graph-like (DAG).

Our first challenge was to determine the ground truth for evaluation. This is due to the fact that the categorization in the dataset was done according to the USPC system, which has been oftenly criticized as complex and unintuitive, with uppermost layer of hierarchy including more than 400 classes, some of them highly redundant (for example, class 131 "Tobacco" and class D27 "Tobacco and smoker's supplies"). After trying to compare the system's output results to the USPC classification and getting very low results (1.8% precision), we decided against using the system's classification, and resolved to use human judgement to understand whether the classification is successful.

The second challenge was to evaluate a complex, evolving hierarchy using a small sample of human observations. We decided to evaluate the classification using pairs of articles that classified either as fit to belong in the same cluster or not; our first intention was to do such manual labeling for two levels of classification, matching the classification structure of IPC (8 major sections and 130 smaller classes); however, we soon understood that the sample size needed to properly evaluate the accuracy of the smaller class division is higher by orders of magnitude than is feasible to evaluate manually.

It is clear to us that the decision to use manual labeling for ground truth severely limited our evaluation capabilities and affected the results; clearly, to evaluate the classification properly, a large-scale experiment with multiple, professional evaluators is needed.

6 Visualization

As we deal with taxonomy and hierarchical clustering, it is natural to choose the dendrogram graph to visualize the resulting hierarchical clusters. Since we cluster millions of patents, a plot of the final dendrogram 'as is' will not be comprehensible; however, this is the main output of the algorithm.

To achieve both comprehensive and accurate visualization, we plotted the main dendrogram, assigning different color to each cluster under a distance of $0.7 * max(linkage)$. In addition, for some of the main clusters we extract the most frequent words of the cluster and highlight this words on the cluster part of the dendrogram. Figure 3 shows the frequent words; it can be seen that the fields of the termins loosely match the major IPC section themes (agriculture, chemistry, electricity, machines etc.)

7 Future Work

There were three challenges in this project that we overcome, but certainly can be improved:

1. More features can be taken into account such as:
   - Advanced textual feature tools like word2vec and NLP tools.
   - Advanced graph analysis tools
2. Richer data - We didn't use the authors of the patent, although it can clearly enrich the feature space as another graph or other features. In addition, further data from different source can be used, such as EU patent datasets, news websites, other legal datasets or 'experts datasets' such as field-specific journals and encyclopedias can be used etc.
3. Feature space - Mixing graph features with text features had to be done carefully to keep the balance and avoid giving too much weight to some features. This part of the project can be done in many different ways and calibrated to specific dataset.
4. Scalability - textual features and community detection of large graphs algorithms are both space and time consuming. Our algorithm can be improved by more sophisticated architecture (e.g. better preprocessing), less access (read&write) to drive and more parallelization.

5. Mixing graph and text feature with more advanced algorithms: For instance, connectivity constraints (citations / authors) can be added to K-means algorithms as connectivity constrains [4]

8 Conclusion

The technical methods of hierarchical clustering is basically simple, the ambitious parts of this project were mixing of graph-clustering and text-clustering, and evaluating such a hazy objective: taxonomy.

Although we used basic features - the results indicated that mixing text and graph features can improve the categorization process. The surprising fact is that the citation information alone can serve as a base for automatic classification of the patents.

We are finishing this project full with new tools and ideas for future work.

References

[1] Raghavan, U.N. and Albert, R. and Kumara, S. Near linear time algorithm to detect community structures in large-scale networks. Phys Rev E 76:036106, 2007. http://arxiv.org/abs/0709.2938.

[2] VD Blondel, J-L Guillaume, R Lambiotte and E Lefebvre: Fast unfolding of community hierarchies in large networks, J Stat Mech P10008 (2008), http://arxiv.org/abs/0803.0476

[3] Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions Halko, et al., 2009 (arXiv:909) http://arxiv.org/pdf/0909.4061

[4] See scikit-learn : adding-connectivity-constraints