Identifying Frauds and Anomalies in Medicare-B Dataset

Jiwon Seo
UNIST, Korea
jseo@unist.ac.kr

Ofer Mendelevitch
LendUp, USA
ofer@lendup.com

***Abstract*— Healthcare industry is growing at a rapid rate to reach a market value of $7 trillion dollars world wide. At the same time, fraud in healthcare is becoming a serious problem, amounting to 5% of the total healthcare spending, or $100 billion dollars each year in US.**

Manually detecting healthcare fraud requires much effort. Recently, machine learning and data mining techniques are applied to automatically detect healthcare frauds. This paper proposes a novel PageRank-based algorithm to detect healthcare frauds and anomalies. We apply the algorithm to Medicare-B dataset, a real-life data with 10 million healthcare insurance claims. The algorithm successfully identifies tens of previously unreported anomalies.

I. Introduction

The amount of healthcare spending in world wide is increasing rapidly to reach $7.2 trillion dollars [7]. In United States, more than $3 trillion dollars is spent on healthcare each year [12]; approximately 5% of the total healthcare spending, or $100 billion dollars each year, is estimated to be lost on healthcare fraud [8]. The problem is similar in EU – 5% of the healthcare budget or $30 to $100 billion dollars is annually lost on fraud.

Detecting healthcare fraud requires significant amount of effort with comprehensive medical knowledge. Traditionally insurance companies rely on a few auditors to manually review and identify suspicious insurance claims. This manual process is expensive and time-consuming. Recent advances of machine learning and data mining techniques led to more efficient and automated detection of healthcare frauds.

In data-mining based healthcare fraud detection, many different algorithms exist; to list some, there are decision tree, genetic algorithm, clustering, and rule-based detection algorithms [5], [8], [14]. It has been studied that the different mining algorithms are effective in detecting different types of frauds [8]. Hence the algorithms are, to some extent, orthogonal to one another and when collectively used the algorithms can detect wide-range of healthcare frauds.

In this paper we propose a new fraud detection algorithm that can be used independently as well as jointly with existing algorithms to detect healthcare frauds and anomalies. Our detection algorithm adopts a variation of PageRank algorithm; our algorithm identifies medical providers whose insurance claims are significantly different from other providers of the same medical specialties. While PageRank is traditionally used in the domain of network analysis, we demonstrated that it can effectively identify anomal insurance claims. We applied the algorithm to Medicare-B dataset, a real-world data containing transactions of insurance claims between 880,000 medical providers and CMS (the United States Center for Medicare and Medicaid Services). It has successfully identified many errors in the dataset, some of which are likely to be fraudulent. We have made public the source code used in this paper in the github repository [2].

II. Background

Medicare-B Dataset is public dataset containing healthcare insurance claim information [11]. The dataset is released into public domain by CMS (the United States Center for Medicare and Medicaid Services) in year 2014. The dataset has insurance claims of medical providers for the treatment they provided; 10 million claims, amounting to $162 billion in total, for 880,000 providers are in the dataset. Table I describes some of the properties of the dataset.

Property	Value
Overall claimed amount	$ 162.211 B
Number of total records	9,153,274
Number of medical providers	880,646
Number of distinct CPT code	5,949
Number of specialties of providers	89

TABLE I
Some Properties of Medicare-B Dataset

The medical treatments are denoted by CPT (Current Procedural Terminology) code [4], which is a set of code (totaling about 9,000 in number) describing tests, surgeries, evaluations, and other medical procedure performed by a medical provider on a patient.

The dataset stores multiple records for each medical provider, and each record describes insurance claims for a single type of medical procedure by the provider. A record contains the medical provider's information, the CPT code for the provided procedure, the number of times the procedure was provided, the claiming amount in dollar, and the number of beneficiaries of the provided procedure. The provider information includes their name, NPI (national Provider ID), specialty, and the location of their services.

In our anomaly detection algorithm, we extract the following four information from each record: provider's NPI, specialty, CPT code, the number of distinct beneficiaries per day. For example, after our extraction process, we can have a record (1005007151, Internal Medicine, 81002, 16), which indicates that the medical provider of ID 1005007151 specializes in Internal Medicine and prescribes CPT code 81002 (Urinalysis) to 16 distinct patients a day on average.

978-1-5090-2809-2/17/$31.00 ©2017 IEEE

Pagerank Algorithm is a link analysis algorithm to compute importance of nodes in a graph [6]. The algorithm computes the stationary probability distribution that a random walker will end up in each node; the nodes with high probability are considered as important. A random walker starts from a randomly selected node; at each step, he either 1) follows one of the edges in the current node to visit next node or 2) he teleports to a randomly selected node.

The probabilities, or PageRank values, can be computed by iteratively applying the following equation:

$$Rank_{t+1}(u) = \frac{1-d}{N} + d \times \sum_{(v,u)\in E} \frac{Rank_t(v)}{degree(v)}$$

That is, the probability for the walker to visit node u at iteration t+1 can be computed by first considering the random teleport factor, then adding up the probabilities of being at adjacent node v at iteration t, multiplied by the probability of following the edge (v, u). The iterative computation is repeated until the PageRank values converge.

There are multiple variations of PageRank algorithm, and one important variation is personalized PageRank algorithm [10]. In personalized PageRank, a random walker starts from a given set of nodes (called seed nodes), instead of starting from or teleporting to a randomly selected node, hence higher priority is given to those selected nodes. In our algorithm, we apply personalized PageRank algorithm to identify anomalies.

III. Anomaly Detection Algorithm

Our algorithm identifies medical providers who prescribed the medical procedures that are significantly different from those of other providers having same specialties. We consider those providers as anomalies that need to be examined for fraudsters. Our assumption is that there would be some similarity in the prescription patterns among the medical providers having same specialties. The algorithm identifies the providers that violate this assumption.

Our anomaly detection algorithm proceeds as following:

1) The similarities between medical providers are computed based on their prescriptions.
2) A similarity graph is generated whose nodes represent providers and edges indicate that the two providers are similar (the similarity is above a given threshold).
3) Then we run personalized PageRank algorithm on the similarity graph. The seed nodes of the algorithm are the medical providers of a selected specialty.
4) We identify as anomalies the providers having high PageRank values but different specialty than the initially selected one.

In the generated similarity graph in step 2, connections among the providers of a same specialty are expected to be denser because the prescriptions are more similar among the providers of a same specialty than among the providers of different specialties. Thus when we run personalized PageRank in step 3 with seed nodes of a selected specialty, PageRank values of the selected specialty are expected to be higher. If a provider of a different specialty has a high PageRank value, that indicates the provider prescribes medical treatments that are normally prescribed by providers of different specialty; in such a case, we better pay attention to the provider and examine if his treatments are legitimate.

Next we describe each step of our anomaly detection algorithm in more detail.

A. Computing Similarity Scores

To compute the similarity scores between medical providers, we create a feature vector for each provider from their overall prescriptions. Each dimension of the feature vector represent a relative frequency of a particular CPT code. Since there are 9,641 CPT code as of year 2012, the total number of the dimensions is 9,641.

More formally the i_{th} dimension of the feature vector v_p of a provider p is defined as, $v_{p,i} = \frac{c_{p,i}}{\sqrt{\sum_i c_{p,i}^2}}$, where $c_{p,i}$ denotes the number of prescriptions of i_{th} CPT code by p.

We define the similarity score between providers as the cosine similarity of their feature vectors [17]. Thus the similarity scores are between 0 and 1.

To generate the similarity graph in the next step, we need to compute the similarity scores of all the pairs of the providers. This requires $O(N^2)$ computation, and with 880,000 providers more than hundred of billions of computations is needed. Since this is too expensive, we only consider the provider pairs that are likely to be similar by applying a simple variant of locality sensitive hashing [9].

For each provider, we examine the feature vector of the provider, and consider the treatments whose dimensions in the vector are larger than a threshold; then the provider is put in the buckets that are designated by those treatments. Then, we only compute the similarities among the providers that are in the same buckets, because similar providers are highly likely to be in the same buckets. For example, if provider p has a feature vector $(v_{p,1}, v_{p,2}, ..., v_{p,k})$ and only two dimensions, or $v_{p,i}$ and $v_{p,j}$, are larger than a threshold, then p and his feature vector is put into two buckets that represent i^{th} and j^{th} CPT codes. Then the provider is compared to other providers that are in those two buckets.

Although the above pruning process works quite well, it is less effective than we expected, because there are a couple of buckets that contains many providers. For example the bucket that contains the providers who frequently prescribed CPT code 99213 (representing "Office/outpatient visit") contains more than 396,000 providers, which is nearly half of all the providers. For such buckets we create smaller sub-buckets that group the providers in terms of their other frequently prescribed treatments. We set the threshold for the sub-buckets be smaller than the threshold for the main buckets because the providers in sub-buckets already have one treatment in common.

B. Generating Similarity Graph

Based on the similarity scores computed in the previous step, we generate the similarity graph of medical providers; the nodes in the graph represent the providers and the edges

indicate that the connected providers are similar in terms of their prescriptions.

We simply connect two providers if their similarity score, or cosine similarity, is higher than a threshold. By default we use the threshold value 0.7; we discuss the threshold sensitivity of the algorithm in Section IV.

C. Running personalized PageRank

Next step in our algorithm is to run personalized PageRank algorithm on the similarity graph. First we select the specialty that we want to examine for the anomaly detection. Then we mark the providers with the selected specialty as seed nodes. With the seed nodes we iteratively run personalized PageRank algorithm.

Since the algorithm starts the random walk from the seed nodes (and teleports to them), and the random walker follows the edges that connects similar nodes, the walker is likely to visit the nodes whose prescriptions are similar to those of the seed nodes. Thus when the algorithm converges, the nodes with the medical prescriptions that are similar to the seed nodes will have high PageRank values.

For faster convergence of the algorithm, we adopt a variant of Pagerank algorithm, named PageRank-nibble [1]. In PageRank-nibble, the random walker stays in a node (hence not following edges) if the node has a very small PageRank value. This variation is known to make a small difference in the final PageRank values, and at the same time makes the algorithm to converge significantly faster.

D. Identifying anomalies

After the PageRank values are computed, we make a list of the providers (and their PageRank values) whose specialties are different from the initially selected one; and we sort the list by their PageRank values.

Then the algorithm iterates over the list and examines the providers and their PageRank values; if it is larger than the smallest PageRank values of the seed nodes, we mark the provider as anomaly. Those providers need to be investigated, because their prescription patterns are very similar to those of providers with different specialty.

Since the list is sorted by the PageRank values, auditors can go over the list and quickly estimate if a provider is likely to be an anomaly/fraudster by comparing his PageRank value against the average/minimum PageRank values of the providers with the selected specialty. Then he can examine the provider's prescriptions and other information in detail to determine if further auditing is required.

The overall anomaly detection process repeats the last two steps for multiple specialties. After the initial preprocessing (first two steps), it allows human auditors to select the specialty to investigate; then it runs the remaining steps of the algorithm to present anomaly providers whose prescriptions are unexpectedly similar to those of the selected providers.

IV. Evaluation

In this section we discuss our experience of applying the anomaly detection algorithm in Medicare-B dataset and present some of the anomalies found by the algorithm.

A. Algorithm Performance

We briefly discuss the performance and resource requirements of the algorithm in this section. We run the algorithm on a single machine having Intel Xeon CPU E5-2640 with six cores running at 2.5GHz frequency; the machine is equipped with 20GB memory but the algorithm uses less than 10GB of the memory to process Medicare B dataset. To implement and run the algorithm, we used SociaLite [15], [16], a main-memory graph processing framework based on Hadoop [3].

The first two steps of the algorithm to generate the similarity graph require offline batch processing, taking less than 1.5 hour. The generated graph is quite large with 673 million number of edges, and the size of the graph varies depending on the threshold value to link (or unlink) a pair of providers. The last two steps of the algorithm run in real-time, taking only 1 to 2 minutes to run. The difference in the running time is due to the difference in the link structures of the selected providers. To further optimize the running time, we use a dictionary encoding that assigns consecutive IDs (starting from zero) to the medical providers; this makes it possible to compactly store the adjacency lists for the graph as well as to promptly index into the offsets of PageRank values and adjacency list of a given provider. This optimization makes the algorithm (last two steps) to run in 10 to 30 seconds.

Specialty	Most Common CPT Code	Cat.
Otolaryngology	92014/Eye exam & treatment 66984/Cataract surg wiol 1 stage	A
Plastic and Reconstructive Surgery	92082/Visual field examination 92285/Eye photography	A
Internal Medicine	17003/Destruct premalg les 2-14 11100/Biopsy skin lesion	B
General Practice	00142/Anesth lens surgery 00400/Anesth skin ext/per/atrunk	B
Internal Medicine	92014/Eye exam & treatment 92136/Ophthalmic biometry	C

TABLE II
SAMPLES OF DETECTED ANOMALIES AND THEIR CATEGORY

B. Anomalies Found

Upon applying the algorithm on Medicare-B dataset, it detected quite a few anomalies; all the anomalies are previously unreported to the best of our knowledge. We have listed some of the anomalies in Table II, which shows the specialties of the anomaly providers, their commonly prescribed CPT code, and the anomaly categories defined as following.

1) Data entry error:

   Many of the detected anomalies seem to be due to the human errors in entering the data. Since the terminologies used in the dataset are rarely used in everyday life, data entry workers seem to have made quite a few mistakes. The most common errors are confusion between ophthalmologist and otolaryngologist (first row in Table II). Many of the medical providers specialized in ophthalmology are recorded as otolaryngologists and vice versa; there exists more than ten errors of this case.

2) Providers with dual or sub specialties:

   Some of the providers in the dataset have dual or sub specialties, and the dataset is missing such information. Those providers with dual/sub specialties often prescribe with their other/sub specialties that are not described in the dataset; hence they are detected as anomalies by the algorithm. We use other datasets such as NPI File [13] to identify the medical providers with dual/sub specialties and put those providers in this category.

   We found quite a few instances of this case, for example, the case of a general practitioner sub-specialized in anesthesiology (the second row from the bottom in Table II). Although majority of the cases in this category are not frauds, some of the cases look suspicious, especially providers with sub specialties who only prescribe with their sub specialties.

3) Unknown error (potentially true fraud)

   Some of the anomalies we found do not fall in the first two categories, nor do they seem to be of other types of false positives. When we discuss the cases with medical doctors, the prescriptions of the medical providers in this category seem rather unnatural.

   Examples in this category include an internist whose prescriptions are similar to those of ophthalmologists. The CPT code that are frequently shown in his prescriptions include "Eye exam & treatment", "Ophthalmic biometry", and "Internal eye photography" (the bottom row in Table II)

C. Discussion

While our algorithm depends on a user selected parameter – the threshold value for similarity scores to connect providers in the similarity graph – the algorithm is not sensitive to the threshold value. We run the algorithm with different threshold values ranging from 0.65 to 0.8, the algorithm gives mostly identical result. The reason why the algorithm is insensitive to the threshold is that the similarity scores are reflected on the link structures of the similarity graph to some extent. If two providers are very similar, then it is likely that the providers have many common neighbor providers in the graph.

When we applied the detection algorithm with different specialties, the algorithm gives more accurate and clear result with more "specialized" specialties than more "general" specialties. That is, if specialties such as otolaryngology, Obstetrics/Gynecology, Dermatology, Cardiology, are selected, the result anomalies seems more clear and accurate. However, when more general specialties such as Certified Nurse Midwife, Nurse Practitioner, Family Practice, Physician Assistant, are selected, the result seem to be somewhat unclear and contain more false positives.

Some of the false positives are due to the "general" characteristics of some specialties. For example, some of the internists are marked as too similar to otolaryngologists because their prescriptions include "Diagnostic laryngoscopy" or "Ear microscopy examination", which are also commonly prescribed by otolaryngologists.

Although we primarily focus on applying the algorithm to directly identify anomalies, it is also possible to use the result of the algorithm in other anomaly detection algorithm. By running the algorithm with different specialties selected, we can create a vector of resulting PageRank values for each providers. The vectors can be used as an input to other training-based anomaly detection algorithms.

V. Conclusion

In this paper we proposed a novel PageRank-based algorithm to identify anomalies/frauds in healthcare data. The detection algorithm is applied to Medicare-B dataset, a real-life data containing information of almost 10 million insurance claims. We have successfully identified tens of anomalies in the dataset and performed in-depth analysis of the detected anomalies.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2016R1C1B1016114).

References

[1] R. Andersen, F. Chung, and K. Lang. Using pagerank to locally partition a graph. *Internet Mathematics*, 4(1):35–64, 2007.
[2] Anomaly detection in medicare-b dataset. https://github.com/ofermend/medicare-demo.
[3] http://hadoop.apache.org.
[4] M. Beebe, J. A. Dalton, M. Espronceda, D. D. Evans, R. L. Glenn, and G. Green. *Current Procedural Terminology: CPT*. American Medical Association, 2007.
[5] R. J. Bolton and D. J. Hand. Statistical fraud detection: A review. *Statistical science*, pages 235–249, 2002.
[6] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *WWW7*, pages 107–117, 1998.
[7] Global health care outlook. https://www2.deloitte.com/content/dam/Deloitte/global/Documents/Life-Sciences-Health-Care/gx-lshc-2015-health-care-outlook-global.pdf, 2015.
[8] J. Gee, M. Button, and G. Brooks. The financial cost of fraud: what data from around the world shows, 2009.
[9] A. Gionis, P. Indyk, R. Motwani, et al. Similarity search in high dimensions via hashing. In *VLDB*, volume 99, pages 518–529, 1999.
[10] T. H. Haveliwala. Topic-sensitive pagerank. In *WWW11*, pages 517–526, 2002.
[11] Medicare dataset – Part B. https://www.cms.gov/Medicare/Medicare-General-Information/MedicareGenInfo/Part-B.html.
[12] National health expenditures 2015 highlights. https://www.cms.gov/research-statistics-data-and-systems/statistics-trends-and-reports/nationalhealthexpenddata/downloads/highlights.pdf.
[13] NPI File. http://download.cms.gov/nppes/NPI_Files.html.
[14] C. Phua, V. Lee, K. Smith, and R. Gayler. A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*, 2010.
[15] J. Seo, S. Guo, and M. S. Lam. SociaLite: Datalog extensions for efficient social network analysis. In *ICDE*, pages 278–289, 2013.
[16] J. Seo, J. Park, J. Shin, and M. S. Lam. Distributed SociaLite: a Datalog-based language for large-scale graph analysis. In *PVLDB*, volume 6, pages 1906–1917, 2013.
[17] P.-N. Tan, M. Steinbach, and V. Kumar. Data mining cluster analysis: basic concepts and algorithms. *Introduction to data mining*, 2013.