Graph-based Diagnostic Medical Decision Support System

Gabriel Bassett
Information Security Analytics LLC
Fairview, TN, USA
gabe@infosecanalytics.com

Kindall Deitmen
College of Computing and Technology
Lipscomb University
Nashville, TN, USA
kindall.deitmen@lipscomb.edu

***Abstract*—Medical diagnosis through differential diagnosis is a matter of life and death that occurs every day. This paper attempts to simulate differential diagnosis using a newly developed machine learning training and query algorithm utilizing a graph-based model. The paper also presents a method for creating synthetic medical records on which to train the model as well as the results of such training.**

Keywords—decision support system, differential diagnosis, graph, machine learning, healthcare

I. Introduction

A. Problem Space

Medical diagnosis is a task that occurs every day and is critical to the life of those being diagnosed. Yet, though it is based on the straight-forward process of differential diagnosis, qualification to make medical diagnoses requires years of education and experience. We propose to build a graph-based model capable of incorporating the large body of existing medical knowledge to automate the execution of differential diagnosis for medical decision support.

Differential diagnosis is a systematic method to determine what is causing symptoms when multiple possibilities exist. A physician will collect the evidence and attempt to discover all the possible causes. Beginning with the most likely causes, he or she will run tests to rule out possible causes until a diagnosis is made. A clinical decision support (CDS) system should "provide clinicians or patients with clinical knowledge and patient-related information intelligently filtered or presented at appropriate times, to enhance patient care" [1].

The need for decision support is so profound that it is a requirement of the U.S. government's Meaningful Use Phase 2 requirements. The Health Information Technology for Economic and Clinical Health Act (HITECH), part of the American Recovery and Reinvestment Act of 2009, incentivizes medical providers to meet certain standards to collect Medicaid and Medicare payments [2]. Healthcare providers were first required to implement electronic health records (EHRs). After getting patient data into an electronic format, providers must show "meaningful use" of EHRs and "begin to realize the true potential . . . to improve the safety, quality, and efficiency of care" [3]. One aspect of meaningful use is to use decision support tools to avoid preventable mistakes and make more effective and efficient decisions.

B. Motivation

Quality decision support could potentially have a significant positive impact on the lives of innumerable people. Decision support can provide medical professionals in underprivileged portions of the world the benefit of aggregate experience far beyond their own. Decision support can also create more consistent diagnoses providing the same outputs given the same inputs. Decision support may also make diagnosis and treatment more efficient by simplifying complex decision processes and extrapolating out potentialities from tests required to treatments offered.

The rest of the paper is organized as follows. Section II describes related work. Section III introduces a synthetic EHR generation system, a graph-based diagnostic Medical Decision Support System (MDSS) model generation algorithm, and a MDSS diagnosis algorithm based on the aforementioned model. Section IV provides results of validation of effectivity of the model. Section V provides the results of validating the model, Section VI notes future work, and Section VII concludes the paper.

II. Related Work

Ginsberg and Offensend wrote about applying decision theory to medical diagnosis in 1968, citing a case study where clinicians used a decision tree model to determine the best next course of action [4]. There was a demonstrable improvement in diagnosis and treatment when the theory was applied to the situation, but this decision tree was manually implemented. With the aid of computers, processing time is lessened and decisions can be made more quickly.

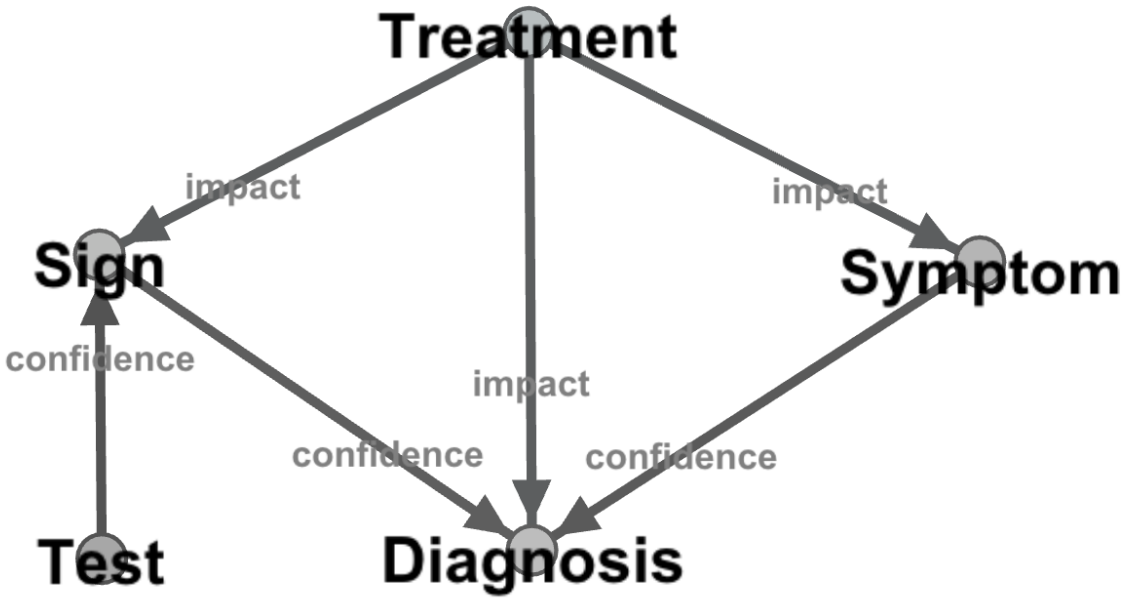

Fig 1. Model Schema

LDS Hospital in Salt Lake City has developed a system called HELP (Health Evaluation through Logical Processing) that supports multiple CDS applications and has EHRs at the center of its design [5]. Clinical information is stored in a common database and contains thousands of medical logic modules (MLMs), which are rules developed by medical professionals in diverse domains to identify certain conditions and treatment options. The EHR information is in a coded format, while transcribed notes are kept as free text. Only the coded information is useful for decision support, but applications have been developed to code some of the free text entries.

Vanderbilt University Medical Center (VUMC) has developed a CDS system using XML tags to organize the CDS rules. [6]. VUMC found their original CDS design was difficult to interpret and did not work well with outside systems and software. By taking their already coded documents, adding XML tags, and presenting diagnosis results on a web page, their CDS rules became more easily understandable. VUMC's schema is designed in such a way that it will be able to accommodate growth using XML's support for extensibility. Because XML is more universal than the previous system, the CDS can interoperate with other systems developed outside of the Vanderbilt system.

St. Jude Children's Research Hospital uses an EHR system with three parts: the Open Clinical Foundation (OCF), PowerChart, and Discern Expert. Open Clinical Foundation is an Oracle database that functions as a data warehouse and stores clinical and administrative information [7]. PowerChart is the user interface that displays the patient's chart to caregivers, stores new test results, and allows the user to see new test results and to indicate which test results he or she reviewed. Discern Expert is the CDS that is built on clinical rules and will generate alerts as well as monitor compliance to treatment plans [7].

Cheng et al. have developed a CDS specifically for use in Intensive Care Units called icuARM. Unlike the previously discussed systems, icuARM provides real-time decision support, which is important for the critically ill patients whose status may be constantly, rapidly changing [8]. The data for icuARM comes from a database with records of more than 40,000 ICU stays of more than 30,000 patients and relies on Association Rules Mining to find connections among these thousands of records. The association rules method allows the system to generate the if-then rules with which it makes its recommendations.

Amato et al. propose Artificial Neural Networks (ANNs) as a method of decision support and diagnosis because of the extensively varied input the models can accept, their ability to find connections that are not readily apparent, and the speed of diagnosis [9]. Support Vector Machines (SVM) have been trained to produce models that aid in CDS. You et al. used SVM to build a model that helped clinicians determine the appropriate dosage for medications to individual patients [10]. Like the ANN, SVM is able to map complex connections that might not be readily apparent to clinicians. Support Vector Machine models offer the added benefit of clearer execution compared to an ANN model.

Bayesian networks have also been proposed to build a CDS and diagnosis engine [11]. Because the medical field has an established ontology with clearly defined terms and concepts, those terms are used to build the basis of an ontology for the CDS. Symptoms are linked to pathologies, and the network gives a probability for each possible connection to the symptoms. As further observations are made and the results added to the algorithm, the probability of a given pathology changes, as do the possible next steps to be taken.

Our system differs from previous systems in its broad scope. The system is not limited in the number of features (signs and

symptoms) it considers or the number of diagnoses it can diagnose. It is not limited by requiring either patient records or Subject Matter Expert (SME) knowledge, but can make use of both sources of information. Additionally, our model provides the ability to integrate tests and treatments that may then be recommended to the user.

Because we do not have access to EHRs to train our model, we developed a system to generate synthetic data that mimics real health records. Buczak et al. of Johns Hopkins University have proposed generating synthetic records to support research that is EHR-based when the researcher does not have access to real, anonymized medical records [12]. Buczak's records were created to look at a population of people with a specific diagnosis, whereas our synthetic records mimic a population with varied diagnoses.

III. A Graph-Based Diagnostic Medical Decision Support System Model Generation Algorithm

A. Model Schema

We chose to model differential diagnosis as a bipartite graph as seen in Figure 1. A bipartite graph is a graph with two classes of nodes in which all edges only relating the two different node classes. Edges never connect nodes of the same class. In this graph, diagnoses will represent one class of node and symptoms[1] and signs[2] a second class of node. Edges in the graph will begin at a symptom or sign and end at a diagnosis the symptom or sign may represent. All edges will be assigned a confidence representing the probability of someone with the symptom or sign having the diagnosis. The confidence is represented as a probability distribution based on the value of the sign or symptom. Throughout the model, signs and symptoms are treated similarly but tracked separately due to the difference in confidence between the two.

The model also allows for the inclusion of test and treatment nodes as seen in Figure 1. These will augment the bipartite graph by providing additional decision support when implemented.

B. Model Training Algorithm

The model training algorithm operates in two phases. In phase one, the records in the training data set are ingested into an intermediate graph. Each record is assumed to contain a single diagnosis and sign/symptom:value key:value pairs. Each sign/symptom is instantiated in the graph as a node and an edge originating at the sign/symptom and ending at the diagnosis created with the value of the sign/symptom stored as an attribute of the edge.

The intermediate graph facilitates the creation of a final graph that is the implementation of the diagnostic MDSS model. Sign/symptom-diagnosis relationships that occur relatively infrequently are assumed to be incorrect relationships and removed from the data. The nature of identifying the cutoff for removal of edges is currently an absolute value. However, the distribution of edges appears as the overlay of a long-tailed distribution representing false sign/symptom-diagnosis relationships and a normal distribution representing true sign/symptom-diagnosis relationships. By calculating a discrete analog of a second derivative, finding the first local minima, and removing all relationships below it, we hope to improve the filtering of relationships. The remaining sign/symptom-diagnosis tuples are then processed. The values of the edges in the intermediary graph between each sign/symptom and diagnosis tuple are modeled as either a Probability Density Function (PDF) or Cumulative Distribution Function (CDF) depending on their nature. A single edge from the sign/symptom to the diagnosis is then stored in the final graph with a value of the chosen density function and its associated characteristics. All density functions are normalized to one for the mean for PDFs and the maximum value for CDFs.

While only a subset of functions were modeled in this step of model generation, a more robust system of distribution fitting is envisioned. Such a system would attempt to fit multiple distributions to the value set, picking the distribution with the minimum error. This improvement can be accomplished without affecting the operation of the rest of the model.

Each edge also contains the number of values that it represents (synonymous with the number of edges between the tuple in the intermediate graph). The in-degree of each diagnosis is also stored on the diagnosis node in the final graph for later use. Finally, each node is labeled to reflect whether it is a sign, symptom, or diagnosis. Both the intermediary graph and the final graph are stored with a list of signs, symptoms, and diagnoses for later reference if necessary.

The current model does not support streaming of incremental updates and must be rebuilt from the intermediary graph in totality. With minimal effort, it is possible however to update only the subset of sign/symptom-diagnosis tuples for which additional data is available. The intermediary graph however may be updated in an incremental fashion.

C. Model Diagnostic Algorithm

A query is initiated with a record comprised of a set of signs/symptoms and values. All potential diagnoses (successor nodes in the graph model to the signs/symptoms within the record) are collected for analysis. Should filtering of the potential diagnoses be desired, it may be implemented at this point to reduce the effort to diagnose the record. However, the current model chooses not to do so.

A score is developed for each remaining potential diagnosis based on the sum of the weighted scores of all signs/symptoms from the record with which the diagnosis has a relationship. The weights used are as follows:

- The value of the probability function stored on the associated edge in the model graph for the given value of the sign/symptom.
- The number of relationships between the sign/symptom and the diagnosis in the intermediate graph relative to the in-degree of the diagnosis node in the intermediate graph.

[1] Subjective evidence of disease or physical disturbance observed by the patient [16]

[2] An objective evidence of disease especially as observed and interpreted by the physician rather than by the patient or lay observer [17]

- A relative weighting representing the confidence difference between signs and symptoms.
- A static weighting associated with the sign or symptom representing its importance relative to other signs and symptoms. For example, finger pain is less important than chest pain. While in the test implementation all weights are set to 1, which allows a means of incorporating SME knowledge. (The other means being through manually added edges).

The sum of these weighted scores for a single diagnosis represents the scores for that diagnosis. The diagnosis:score tuples represent the output of the diagnostic algorithm. The scores reflect the relative confidence the system has that an individual diagnosis is correct. The largest score is the diagnosis the system believes is most likely to be correct. An example of output of the diagnostic algorithm may be found in Appendix 1.

D. Technical Implementation

The model and all algorithms are implemented in the Python programming language [3] stored in memory. The networkx python module[4] is used to represent graph objects while the scipy python module [5] is used to represent distributions. The implementation is accessed through standard web protocols either using a web browser on a computer or mobile device, or an application programming interface for machine-to-machine interactions. The web interface is implemented using the python flask module[6].

The implementation should function correctly on any system that supports the Python language including Windows, Linux, and Apple OS X. The primary constraint is expected to be system resources including random access memory, processing speed, and storage. The implementation does not support parallel execution due to lack of support in the utilized modules. Load-balancing implementations are likely possible but not explored.

The use of an in-memory graph object represents a change from the Neo4j [7] graph database envisioned in the original conceptualization of the model. This is due to the storage of distribution objects on edges preventing serialization of the graph. Future improvements may modify the implementation to store the distribution characteristics rather than the distribution itself. This would allow serialization of graph attributes to facilitate storage of the graph in a graph database or flat file.

E. Graphical User Interface Implementation

The Graphical User Interface (GUI) is implemented in JavaScript (JS), Cascading Style Sheets (CSS), and Hypertext Markup Language (HTML). It presents the user with a row in which to place a sign or symptom name and its value. Buttons are provided to add and remove rows for additional diagnoses.

A diagnose button is provided to execute the diagnosis. The signs/symptoms are sent to the server, diagnosed, and the resulting diagnoses/scores returned. The resulting diagnoses are presented to the user in a table in order of their score (from highest to lowest). The scores are presented as a percentage difference from the top score such that the top score is 0 and the minimum possible score is -100%.

The list of diagnoses is limited to the top five. This is an arbitrary limited. Additional diagnoses could be included at any set number or upon request through the GUI by the user.

Diagnoses could additionally be paired with their prevalence in the general population, allowing a user to sort on the most common diagnoses and see their score relative to the top diagnoses. Diagnoses could also be paired with their criticality so that they may be sorted by such, ensuring that critical but unlikely diagnoses are not overlooked.

As part of a future update to the GUI, tests that may be run to differentiate between the diagnoses will be suggested and treatments to address the top diagnoses will be recommended.

F. Application Programming Interface

The model implementation exposes three functions through an application programming interface. The first receives signs/symptoms as key:value pairs in a serialized JavaScript Object Notation (JSON) string. The diagnoses and scores are returned in a similar format as a list of tuples in a JSON string. The ability to generate synthetic records is also exposed. The number of records requested is passed through the API and the records are returned as a list of dictionaries in a JSON string. Finally, the ability to query the truth data is exposed. A diagnosis may be passed to the API and a dictionary containing the truth diagnosis and the associated signs/symptoms as well as their values are returned as a JSON string.

IV. Model Generation Example using Synthetic Data

A. Synthetic Data Creation Methodology

The model is designed to utilize EHRs for both training and diagnostic purposes. As no EHR stores of sufficient quantity and quality were available within the existing constraints, a system for generating synthetic EHRs was created. The system produces truth data as well as EHRs as described below.

The synthetic data is based, in part, on static values identified through consultation with an SME. Multiple values necessary for generating the synthetic data were identified. See Appendix 2 for a complete list of static values.

1) Synthetic Truth Data Creation

Generation of synthetic truth data begins with the creation of the desired number of signs, symptoms, and diagnoses. Each sign/symptom is designated as either a continuous or discrete distribution for its values with a given probability and then randomly assigned one of five distribution types:

- Continuous
  - Gaussian

[3] https://www.python.org/
[4] http://networkx.github.io/
[5] http://docs.scipy.org/doc/scipy/reference/stats.html
[6] http://flask.pocoo.org/
[7] http://neo4j.com/ [15]

- Gaussian Cumulative Density Function
- Discrete
  - 10 step – (i.e., What is your pain on a scale from 1 to 10?)
  - 3 step – yes = 1, maybe = .5, no = -1
  - 2 step – yes/no

Once distributions have been assigned to the signs and symptoms, each diagnosis is assigned a number of signs and symptoms based on a Gaussian distribution. The number of signs/symptoms designated are picked from the pool of signs and symptoms and assigned to that diagnosis along with an associated value picked from a pool of potential values. The assignment of signs/symptoms to diagnoses may be preferential in nature with some signs/symptoms being picked significantly more often than others. For our analysis, signs were picked preferentially and symptoms were not. A synthetic truth data example may be found in Appendix 1.

2) Synthetic Diagnostic Records Creation

Once a truth data set has been generated, any number of EHRs may be generated from it. For each requested EHR, a diagnosis is picked. A number of signs/symptoms is picked based on the defined distributions of signs and symptoms for EHRs. Based on the defined probabilities, the signs and symptoms are either chosen from those associated with the diagnosis or from the set of all signs/symptoms. Once signs and symptoms have been assigned, their values are picked from their value distribution such that the defined percentage of signs/symptoms are outliers. This is repeated until the requested number of EHRs has been generated. An example of a synthetic record may be found in Appendix 1.

V. Model Validation

A. Model Training Methodology

To validate the functionality of our model, we began by generating truth data based on the ground-truth variables found in Appendix 2. We then generated multiple sets of synthetic records with sizes from 100,000 to 5,000,000. All records were based on the same truth data and were additive (i.e., the 200,000 record set included the 100,000 record set records plus 100,000 additional new records). A new intermediate graph and final model graph were then created off of each set of training records.

B. Model Validation Methodology

For each model generated in the above training methodology, we generated 1000 new and unique test records that were only used to assess that specific model. The model was used to diagnose all 1000 records and the results compared to the correct diagnosis provided by the record generation algorithm. For each record set, four values were captured:

- Was the correct diagnosis the highest scoring diagnosis
- Was the correct diagnosis in the top five highest-scoring diagnoses
- What was the correct diagnosis score relative to the top scoring diagnosis
- What was the position in the ordered list of diagnoses of the correct diagnosis

C. Results

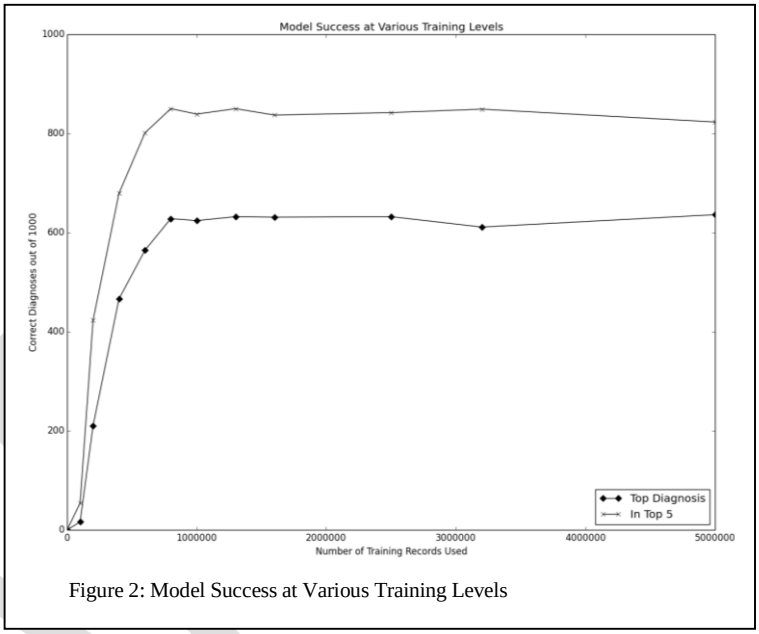

Figure 2: Model Success at Various Training Levels

As seen in Figure 2, the model improves significantly with additional training records until roughly 800,000. At this point, it plateaus with the correct diagnosis being the top scoring in roughly 65% of test records and in the top five diagnoses in 85% of records. At 800,000 training records, the correct diagnosis is in the top 10 scoring diagnoses over 90% of the time. All models trained with greater than 600,000 records contain the true diagnosis in the top 20 90%+ of the time with the greatest being 800,000 at 94.1%.

To analyze all diagnoses, not just those in the list of the top five, Figure 4 provides histograms of the location of the top diagnosis in a list of all potential diagnoses sorted by score for various levels of training. Subplots 100,000 and 200,000 show additional records in the first bin as true diagnoses not in the potential diagnoses list received a value of -1. Overall, we can see a clear trend. After 600,000 training records, the true diagnosis is in the top bin the vast majority of the time. Figure 5 provides an expansion of all but the first two subplots of Figure 4. In it, we can see the vast majority of records containing the true diagnosis in the top 20 scores.

In Figure 6 we look at the difference between the top score and the true diagnosis score. After the model becomes accurately trained, the distribution of percent difference of scores between the top score and the true diagnosis score is fairly evenly distributed. The spike in the first bin is due to the true diagnosis being the top-scoring diagnosis and the difference, therefore, being zero.

This even distribution does not affect the ranking of the true diagnosis though, as can be inferred from Figures 4 and 5 and seen in Figure 7. Regardless of a large relative difference, the

top score still appears near the top of scores of potential diagnoses.

D. Analysis of Results

It appears that the model succeeds in predicting the correct diagnosis at the top or near the top of list of potential diagnoses a significant amount of the time. While not accurate enough to make definitive final diagnoses, it does appear accurate enough to assist in multiple areas:

- provide support to those making medical diagnoses
- suggest treatments based on the top 20 scoring diagnoses
- suggest tests that would differentiate among the top 20 scoring diagnoses

One concern is the number of records necessary for training. The model clearly benefits from additional records until some point between 800,000 and 1,000,000 records. This is significantly more than a single medical practitioner will see in a lifetime. As such, the model will require a store of EMRs that span multiple medical practitioners over a long time period.

VI. Future Work

This model is the first step in a system for differential diagnosis through machine assistance and can be improved in multiple ways to yield improvements in results.

We intend to implement two additional types of nodes in the graph model: tests and treatments. Tests will represent anything that can produce a sign. Edges will be added from tests to signs indicating that the test produces a value for the sign. The diagnosis algorithm will then be updated to recommend tests that will produce signs that may differentiate between the top scoring diagnoses.

Treatments are actions that can improve medical outcomes. Edges will be added to the model from a treatment to diagnoses that are affected by the treatment, either positively or negatively. The edge will carry an 'impact' property that indicates both the type of impact (positive or negative) and magnitude of impact. The diagnosis algorithm will then provide potential treatments prioritized by their potential positive impact on the top diagnoses while minimizing their negative impact. The addition of these edges does not affect the bipartite nature of the graph with respect to diagnosis and symptom/sign nodes.

As outlined above, allowing the user to expand the number of potential diagnoses she sees and to sort the data by how common the diagnosis is in the population and/or the severity of the diagnosis' impact provides another opportunity for enhancing the model

There are several other theories and models that could benefit the MDSS. Analysis of Completing Hypotheses [13] may be useful to improve the diagnostic process. Replacing the graph layer between the signs/symptoms and the diagnosis with another machine learning model such as a neural network may also improve its accuracy. The method for fitting distributions to sign/symptom values could be replaced with a more robust model able to more accurately match a larger number of distributions. The method for estimation of scale of the distribution of edges in the intermediate graph could be improved to provide a more accurate cutoff between true and false sign/symptom->diagnosis relationships.

While the diagnostic algorithm is based on relative confidence, a logical next step would be to allow signs/symptoms to be marked as required in the input. This would eliminate any diagnosis without an existing relationship to the sign/symptom. While this may cause incorrect information to unfairly influence the diagnosis, it would provide a means of firmly basing a diagnosis off of a high-confidence sign or symptom.

Finally, by aligning the model to standard medical ontologies such as ICD-10 classifications [14], greater interoperability may be obtained. While this does not affect the prediction capabilities of the model, it would facilitate integration of the model into other systems. Simply training the model on actual EHRs may produce the desired effect on support for integration as the EHRs would inherently contain ICD-10 records.

VII. Conclusion

We successfully created a model and user interface for providing clinical decision support and aiding in differential diagnosis. The web interface makes the system accessible to professionals around the world in a user-friendly manner. There are many improvements to be made; however, this represents a significant step forward in differential diagnosis through machine assistance; an accomplishment of significant potential to benefit humanity.

Acknowledgment

The authors wish to thank Jasmine Henry who contributed significant material and moral support, Jeff Crawford for bringing us together, and Eric Stephens for providing great guidance.

VIII. References

[1] J. e. a. Osheroff, Improving outcomes with clinical decision support: an implementor's guide, Productivity Press, 2005.

[2] D. Blumenthal, "Launching HITECH," *New England Journal of Medicine,* vol. 362, 2010.

[3] D. a. M. T. Blumenthal, "The "Meaningful Use" Regulation for Electronic Health Records," *New England Journal of Medicine,* vol. 363, 2010.

[4] A. S. Ginsberg and F. L. Offensend, "An Application of Decision Theory to a Medical Diagnosis-Treatment Problem," The RAND Corporation, Santa Monica, 1968.

[5] R. A. Greenes, Clinical Decision Support: The Road Ahead, Amsterdam: Elsevier Academic Press, 2007.

[6] L. Wing, "Clinical knowledge management: Using Extensible Markup Language to characterize clinical decision support," *American Journal of Health-System Pharmacy*, vol. 71, no. 12, 2014.

[7] K. e. a. Fonkych, The State and Pattern of Health Information Technology Adoption, Santa Monica: RandCorp, 2005.

[8] C.-W. e. a. Cheng, "icuARM - An ICU Clinical Decision Support System Using Association Rule Mining," vol. 1, 2013.

[9] F. e. a. Amato, "Artificial neural networks in medical diagnosis," *Journal of Applied Biomedicine,* vol. 11, pp. 47-58, 2013.

[10] W. e. a. You, "Personalized Drug Administrations Using Support Vector Machine," in *Biomedical Engineering and Informatics*, Shanghai, 2011.

[11] G. Bucci, V. Sandrucci and E. Vicario, "Ontologies and Bayesian Networks in Medical Diagnosis," in *Hawaii International Conference on System Sciences*, Kauai, 2011.

[12] A. e. a. Buczak, "Data-driven approach for creating synthetic electronic medical records," *BMC Medical Informatics and Decision Making,* vol. 10, no. 1, 2010.

[13] R. J. H. Jr, "Chapter 8: Analysis of Competing Hypotheses," in *Psychology of Intelligence Analysis*, Center for the Study of Intelligence, Central Intelligence Agency, 1999.

[14] World Health Organization, "International Classification of Diseases (ICD)," World Health Organization, 2010. [Online]. Available: http://www.who.int/classifications/icd/en/. [Accessed 03 03 2015].

[15] Neo Technology, Inc., "The Neoj Manual v2.1.6 15.5. Capacity," Neo4j, 27 11 2014. [Online]. Available: http://neo4j.com/docs/stable/capabilities-capacity.html. [Accessed 29 11 2014].

[16] Merriam-Webster, Incorporated, "Symptom," Merriam-Webster Incorporated, [Online]. Available: http://www.merriam-webster.com/medical/symptoms. [Accessed 29 11 2014].

[17] Merriam-Webster, Incorporated, "Sign," Merriam-Webster, Incorporated, [Online]. Available: http://www.merriam-webster.com/medical/sign. [Accessed 29 11 2014].

FIG. 3 EXAMPLE MODEL

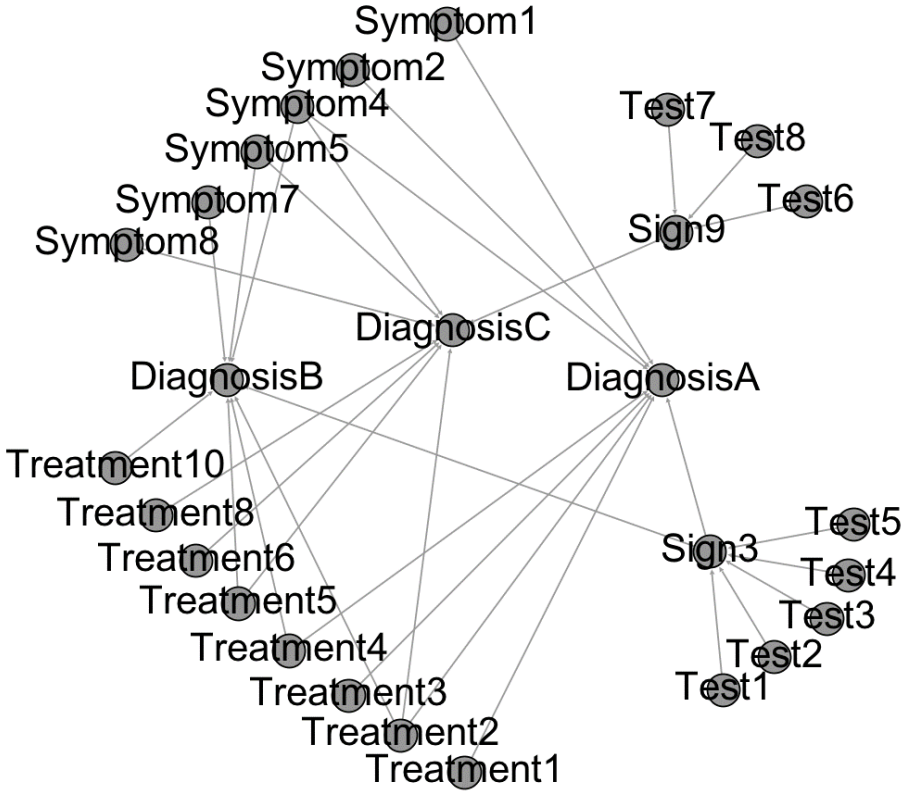

Figure 4

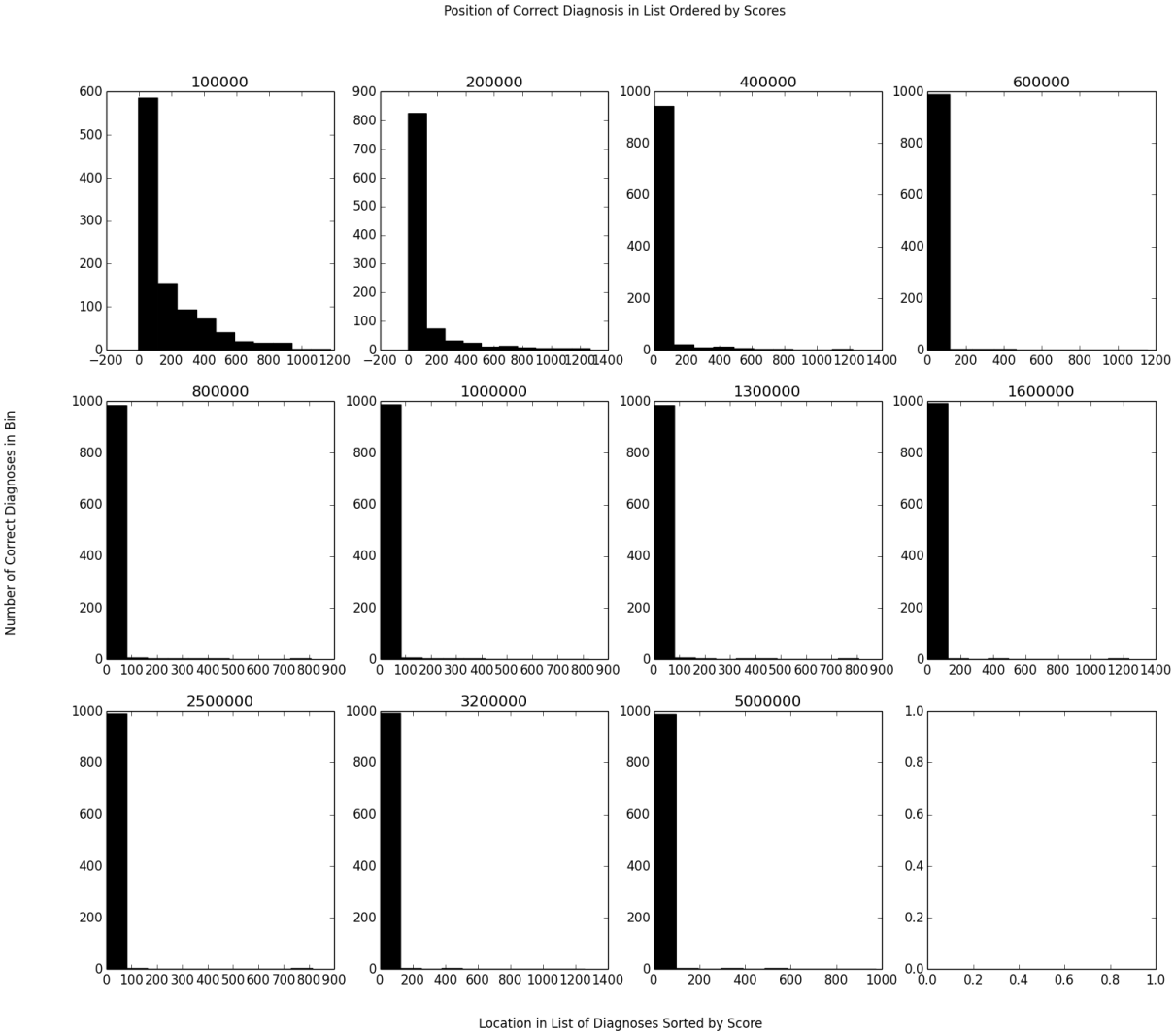
Position of Correct Diagnosis in List Ordered by Scores
100000
200000
400000
600000
800000
1000000
1300000
1600000
2500000
3200000
5000000
Number of Correct Diagnoses in Bin
Location in List of Diagnoses Sorted by Score

Figure 5

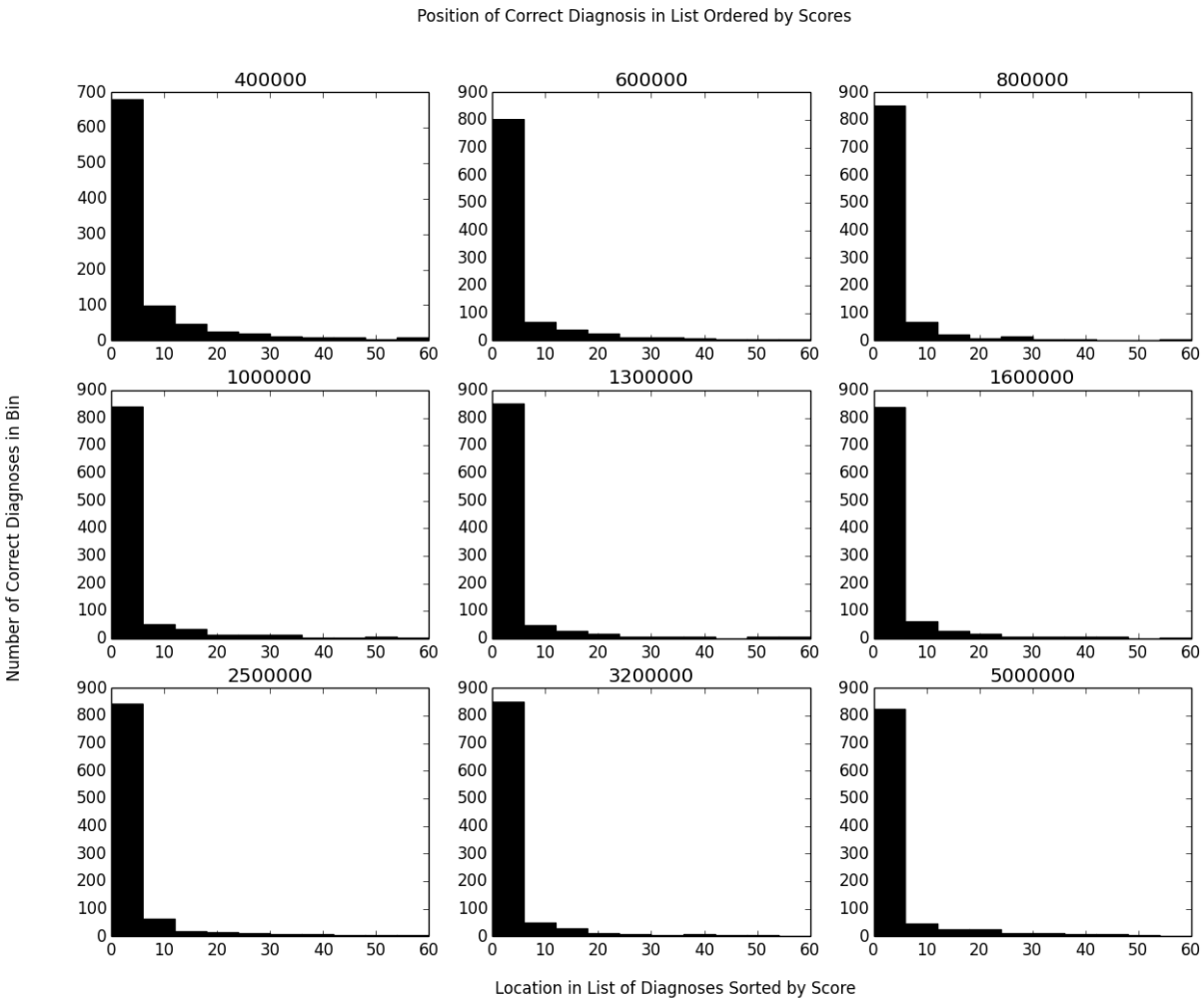
Position of Correct Diagnosis in List Ordered by Scores
400000
600000
800000
1000000
1300000
1600000
2500000
3200000
5000000
Number of Correct Diagnoses in Bin
Location in List of Diagnoses Sorted by Score

Figure 6

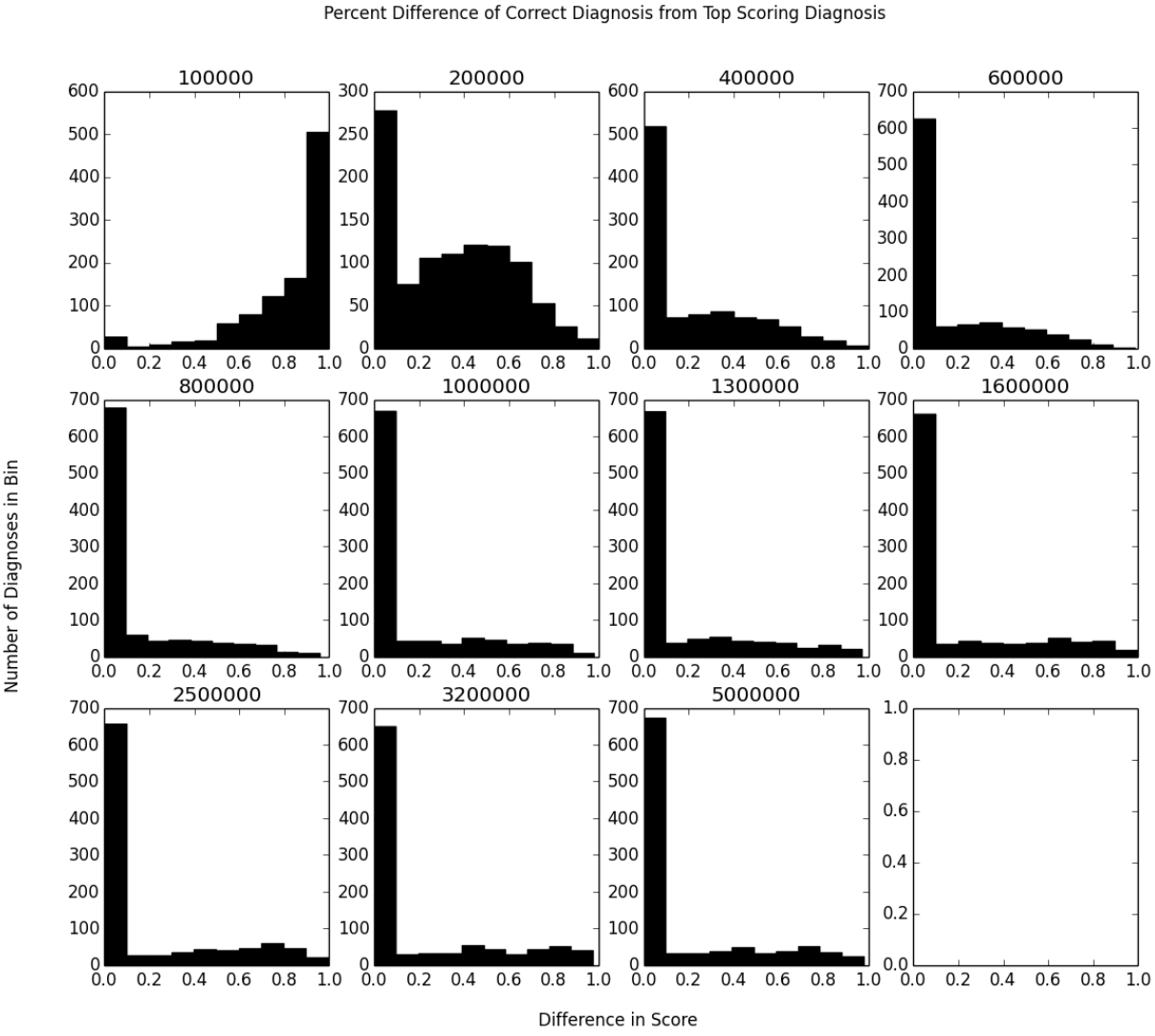
Percent Difference of Correct Diagnosis from Top Scoring Diagnosis
100000
200000
400000
600000
800000
1000000
1300000
1600000
2500000
3200000
5000000
Number of Diagnoses in Bin
Difference in Score

Figure 7

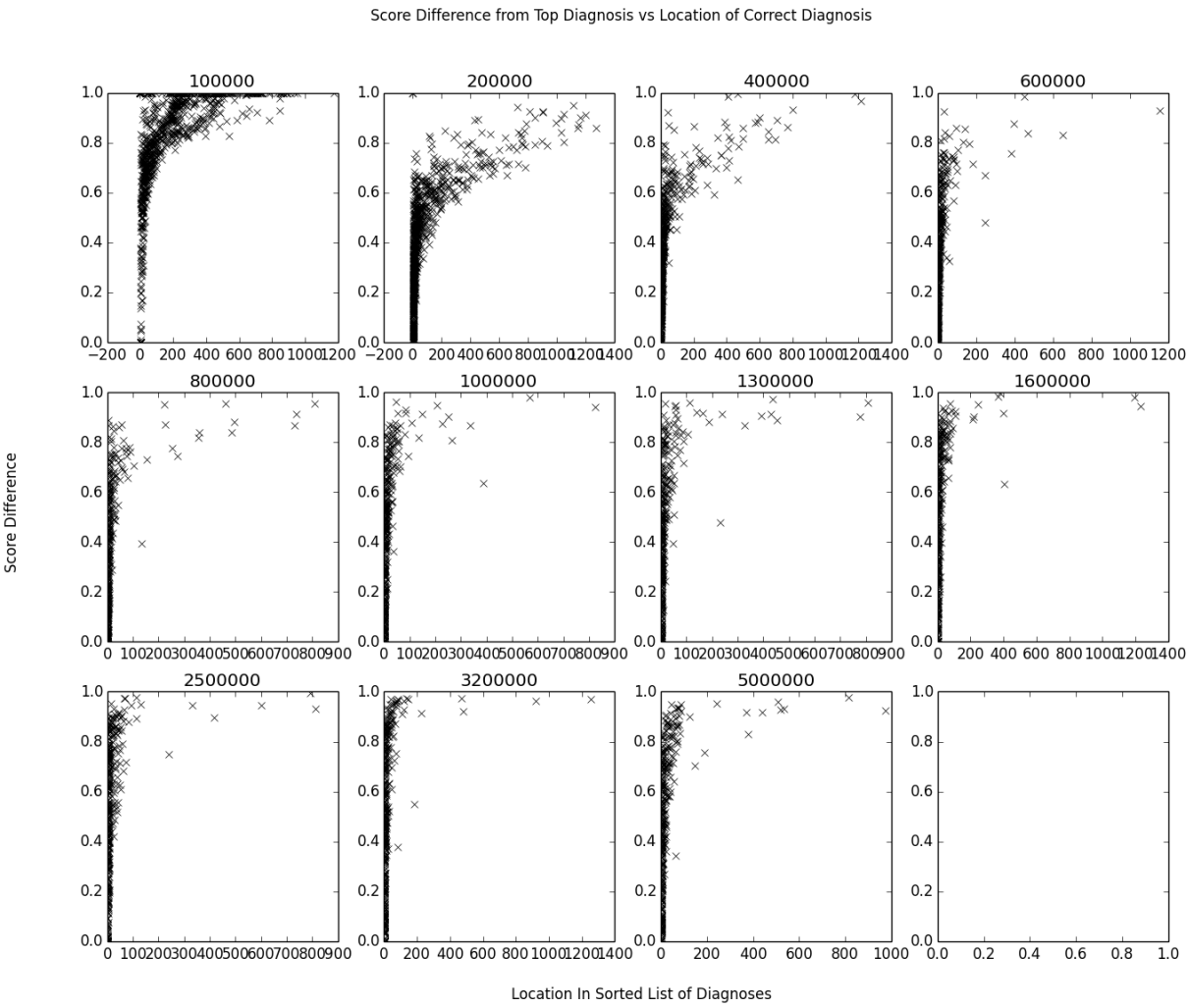
Score Difference from Top Diagnosis vs Location of Correct Diagnosis
100000
200000
400000
600000
800000
1000000
1300000
1600000
2500000
3200000
5000000
Score Difference
Location In Sorted List of Diagnoses

TABLE 1. EXAMPLE MODEL EDGE TABLE

Source	Destination	Impact	Confidence
Sign3	DiagnosisB		f(x) = .95*x
Sign3	DiagnosisA		f(x) = .86*x
Sign9	DiagnosisC		f(x) = 1*x
Treatment8	DiagnosisC	1	
Symptom2	DiagnosisA		f(x) = 1|x==True
Symptom1	DiagnosisA		f(x) = 1/(1*sqrt(2(pi))*e^-1((x-0)^2/(2*1^2))
Treatment10	DiagnosisB	-1	
Symptom5	DiagnosisC		f(x) = 1/x^2
Symptom5	DiagnosisB		f(x) = x
Symptom4	DiagnosisC		f(x) = .95|x==True
Symptom4	DiagnosisB		f(x) = 4^x*e^-4/x!
Symptom4	DiagnosisA		f(x) = x
Symptom8	DiagnosisC		f(x) = x/n
Test1	Sign3		f(x) = x
Test3	Sign3		f(x) = x
Test2	Sign3		f(x) = x
Test5	Sign3		f(x) = x
Test4	Sign3		f(x) = x
Test7	Sign9		f(x) = x
Test6	Sign9		f(x) = x
Treatment6	DiagnosisC	1	
Test8	Sign9		f(x) = x
Treatment4	DiagnosisB	1	
Treatment4	DiagnosisA	-1	
Treatment5	DiagnosisC	1	
Treatment5	DiagnosisB	1	
Treatment2	DiagnosisC	-1	
Treatment2	DiagnosisB	1	
Treatment2	DiagnosisA	1	
Treatment3	DiagnosisA	1	
Treatment1	DiagnosisA	1	
Symptom7	DiagnosisB		f(x) = x

APPENDIX 1

Example Synthetic Truth Data:


```
{'diagnosis_6827': {'signs': {'sign_1534': {'factors': {'inverse': True},
                                            'function': 'bool',
                                            'function_type': 'categorical'},
                              'sign_1939': {'factors': {'inverse': True},
                                            'function': 'bool',
                                            'function_type': 'categorical'},
                              'sign_2345': {'factors': {'inverse': False},
                                            'function': 'bool',
                                            'function_type': 'categorical'},
                              'sign_59': {'factors': {'levels': [0.1,
                                                                 0.2,
                                                                 0.3,
                                                                 0.4,
                                                                 0.5,
                                                                 0.6,
                                                                 0.7,
                                                                 0.8,
                                                                 0.9,
                                                                 1]},
                                          'function': 'step_10',
                                          'function_type': 'categorical'},
                              'sign_982': {'factors': {'levels': [1,
                                                                  0.9,
                                                                  0.8,
                                                                  0.7,
                                                                  0.6,
                                                                  0.5,
                                                                  0.4,
                                                                  0.3,
                                                                  0.2,
                                                                  0.1]},
                                           'function': 'step_10',
                                           'function_type': 'categorical'}},
               'symptoms': {'symptom_112': {'factors': {'inverse': True},
                                            'function': 'bool',
                                            'function_type': 'categorical'},
                            'symptom_134': {'factors': {'inverse': True},
                                            'function': 'bool',
                                            'function_type': 'categorical'},
                            'symptom_25': {'factors': {'levels': [1,
                                                                  0.5,
                                                                  -1]},
                                           'function': 'step_3',
                                           'function_type': 'categorical'},
                            'symptom_49': {'factors': {'levels': [1,
                                                                  0.5,
                                                                  -1]},
                                           'function': 'step_3',
                                           'function_type': 'categorical'},
                            'symptom_78': {'factors': {'levels': [0.1,
                                                                  0.2,
                                                                  0.3,
                                                                  0.4,
                                                                  0.5,
                                                                  0.6,
```

```
                                        0.7,
                                        0.8,
                                        0.9,
                                        1]},
                            'function': 'step_10',
                            'function_type': 'categorical'},
                'symptom_89': {'factors': {'levels': [1,
                                        0.5,
                                        -1]},
                            'function': 'step_3',
                            'function_type': 'categorical'},
                'symptom_97': {'factors': {'levels': [0.1,
                                        0.2,
                                        0.3,
                                        0.4,
                                        0.5,
                                        0.6,
                                        0.7,
                                        0.8,
                                        0.9,
                                        1]},
                            'function': 'step_10',
                            'function_type': 'categorical'}}}}
```


Example Synthetic Record:


```
[{'diagnosis': 'diagnosis_255',
  'signs': {'sign_1404': 0.1},
  'symptoms': {'symptom_11': 0.5,
               'symptom_24': 0.3,
               'symptom_72': 1,
               'symptom_80': 0}},
 {'diagnosis': 'diagnosis_4789',
  'signs': {'sign_1071': 0, 'sign_2259': 0.4},
  'symptoms': {'symptom_12': 1,
               'symptom_135': 0.5,
               'symptom_34': 0,
               'symptom_40': -0.22229441461509403,
               'symptom_5': 0.20000000000000001,
               'symptom_96': 0}},
 {'diagnosis': 'diagnosis_4124',
  'signs': {'sign_382': 0.8, 'sign_584': 0.5},
  'symptoms': {'symptom_30': 1,
               'symptom_43': 0.1,
               'symptom_46': 1,
               'symptom_55': 0.5}},
 {'diagnosis': 'diagnosis_8354',
  'signs': {'sign_1632': 0.2, 'sign_2924': 0.8},
  'symptoms': {'symptom_130': 1,
               'symptom_33': 0,
               'symptom_54': 1,
               'symptom_69': 0,
               'symptom_96': 1}}]
```


Example Diagnostic Result:


```
{'diagnosis_2287': 0.03583184375312342,
 'diagnosis_2497': 0.03916503852085583,
 'diagnosis_2635': 0.86711857628206257,
```

'diagnosis_2789': 0.75093602194361631,
'diagnosis_2964': 0.97376952416702212,
'diagnosis_3043': 0.60280970543672796,
'diagnosis_357': 0.046664726748253761,
'diagnosis_3643': 1.3094644335827743,
'diagnosis_4580': 36.165387964903829,
'diagnosis_4710': 1.9590948408926703,
'diagnosis_4983': 1.6087705898302658,
'diagnosis_520': 0.039720570982144564,
'diagnosis_5431': 8.0418490804417981,
'diagnosis_5794': 0.026088659047444063,
'diagnosis_7509': 0.664636341891777,
'diagnosis_7676': 0.94285620593949759,
'diagnosis_7791': 0.72646297834682605,
'diagnosis_8460': 0.64917968277801474,
'diagnosis_8674': 1.755022461633017,
'diagnosis_8680': 0.040831635904722031
...}

Appendix 2

- Number of diagnoses: 10,000
- Number of signs: 3,00
- Number of symptoms: 150
- Mean signs per diagnosis: 5.5
    - The signs per diagnosis defines a distribution of how many signs each diagnosis will have in the truth data
- Standard Deviation (SD) of signs per diagnosis: 0.5
- Mean symptoms per diagnosis: 7
    - The symptoms per diagnosis defines a distribution of how many symptoms each diagnosis will have in the truth data
- SD of symptoms per diagnosis: 0.5
- Percent of signs represented by continuous functions: 5%
    - An example of a continuous sign would be patient temperature
- Percent of signs represented by discrete functions: 95%
    - An example of a discrete function would be testing positive or negative on a given test
- Percent of symptoms represented by continuous functions: 5%
    - An example of a continuous symptom may be how long the patient had been feeling ill
- Percent of symptoms represented by discrete functions: 95%
    - An example of a discrete function would be a patient's opinion of how bad their pain was on a scale of one to ten
- Preferential attachment of signs: true
    - Represents whether all signs will be equally likely or whether some signs occur very commonly and some occur rarely
- Preferential attachment of symptoms: false
    - Represents whether all symptoms will be equally likely or whether some symptoms occur very commonly and some occur rarely
- Mean signs per record: 2.5
    - The signs per record defines a distribution of how many signs each record will have in the synthetic records
- SD of signs per record: 0.3
- Mean symptoms per record:
    - The symptoms per record defines a distribution of how many symptoms each record will have in the synthetic records
- Percentage of false signs: 8%
    - The percentage of false signs represents two values. The first is the likelihood that a sign in a record will be drawn from all signs rather than just those associated with the record's correct diagnosis in the truth data. It also represents the likelihood that the value for the sign will be an outlier in the distribution that represents the values for the sign.
- Percentage of false symptoms: 20%
    - The percentage of false symptoms represents two values. The first is the likelihood that a symptom in a record will be drawn from all symptoms rather than just those associated with the record's correct diagnosis in the truth data. It also represents the likelihood that the value for the symptom will be an outlier in the distribution that represents the values for the symptom.