Type 2 Diabetes Mellitus Classification

Ayush Sood, Steven Diamond, Shizhi Wang
Department of Computer Science, Stanford University
(Dated: December 14, 2012)

We consider the problem of predicting whether a given patient has Type 2 Diabetes Mellitus using his or her electronic health records (EHR), which often lack common indicators of diabetes. Effective heuristics such as FINDRISC exist for detecting undiagnosed diabetes, but require that patients be screened to collect all necessary information. We compare FINDRISC, restricted to the information in our data set, with models of varying complexity. We note that an ensemble of random forest and gradient boosting machine models vastly outperforms FINDRISC. Furthermore, the ensemble's performance on our data is competitive with FINDRISC's performance in the field. We conclude that diabetes can be effectively detected using EHR alone. Computer diagnosis can, thus, complement more expensive screening.

I. Introduction

Type 2 Diabetes Mellitus is a metabolic disorder that is defined by high blood glucose levels due to insulin resistance and a relative insulin deficiency. Diabetes affects 347 million people worldwide [1]. If current trends continue, the CDC estimates 1 in 3 U.S. adults could have diabetes by 2050 [2]. Type 2 diabetes is responsible for 90% of all diabetes cases [1]. Hereafter we will abbreviate type 2 diabetes as diabetes.

Despite the dangers of leaving diabetes untreated, the WHO does not recommend universal screening because of the expense [3]. Diagnosing diabetes costs up to $253 in lab tests [4]. Many countries screen for diabetes using cheaper heuristics and only do lab tests on likely cases [5]. FINDRISC is one of the most widely used heuristics [5]. Though collecting the data needed by FINDRISC is simple, it still requires a screening session with patients. This not only involves patient cooperation but also the cost of a doctor visit.

We investigate whether undiagnosed diabetes can be detected reliably from EHR, without screening patients for diabetes specific risk factors. If this problem were solved, medical providers could automatically search for likely diabetes cases in their existing records. Such computer diagnosis would cheaply and effectively improve programs to find and treat diabetics.

II. Data

Our data set was provided by Kaggle as part of the "Practice Fusion Diabetes Classification" challenge [6]. The data set contains anonymized medical records for over 10,000 patients, with labels indicating whether or not each patient was diagnosed with diabetes. We divided the data set into test and training sets through stratified sampling.

All obvious indicators of diabetes besides the diagnosis, such as blood glucose levels, were removed from the data. The data set also lacks many features used by FINDRISC and other common screening heuristics, such as patient diet, physical activity, and family medical history. The data set thus simulates the EHR of a medical provider that has not screened patients specifically for diabetes.

A. ICD9 Codes

The data set included a diagnosis history for each patient. These diagnoses were represented using the ICD9 encoding scheme [7]. The scheme has a four level hierarchical structure, where specific diagnoses fall under broader classes of conditions [8]. For example, the diagnosis "essential hypertension" falls under "hypertension" and "diseases of the circulatory system". We generated features not only for each ICD9 code that appeared in our data set but also for the higher level conditions that the code fell under.

B. NDC Codes

The data set had records of all the medications each patient was taking. These medications were identified by their "NDC Codes", which are unique product identifiers for drugs [9]. To get useful data out of the NDC Codes, it was important to map them to the conditions that they treated. That is because there are many drugs that treat the same underlying condition - thus we must cluster them together.

To tackle this issue we mapped each one of the NDC Codes to the active principle ingredient in the drug and then mapped each one of those active principle ingredients to the underlying condition it treated. For example, the NDC Code 247112960 refers to a drug "Mevacor", whose active principle ingredient is "Lovastatin", which is a chemical used to treat "High Cholesterol". Thus we were able to map $\sim 20,000$ NDC Codes into a condensed 20 dimensional feature set.

C. General Health

For each patient, we were given a full record of every single doctor visit they had between the years 2009 and 2012. Each doctor visit provided us with basic information such as weight, height, BMI, systolic blood pressure, and diastolic blood pressure. The provided data was very noisy; therefore, each visit was sanity checked and cleaned. More specifically, all outliers were removed and the BMI was recalculated.

Lastly, we created features to capture the frequency of doctor visits and the types of medical specialists visited, lab tests, and general information such as age and sex.

D. Pre-Processing

Because the data set was highly skewed, with non-diabetic patients outnumbering diabetics by almost 4 to 1, we duplicated each diabetic patient's records 3 times in the training set. We also normalized all continuous features.

III. Performance Metric

Accuracy is a natural metric to use for evaluating models in a classification problem. But since our data was skewed, optimizing for accuracy tended to produce models that classified all patients as non-diabetic. We instead used area under the Receiver Operating Characteristic (ROC) curve, or AUC, as our performance metric. AUC is superior to other metrics that balance precision and recall, such as F-score, because it does not commit to a particular probability threshold for classifying patients as diabetic.

IV. Methods

We used FINDRISC, restricted to our feature set, as a baseline model. We compared the baseline with a range of increasingly complex models. We first trained our models on the full feature set described in II. We then constructed a condensed feature set with the top 10 features. These were the 10 features most correlated with diabetes, skipping several hypertension features that were near duplicates of the most correlated hypertension feature. We trained all the models on the top 10 features.

For each model, after selecting hyper parameters, we determined a 95% confidence interval for the AUC. Specifically, we selected 10% of the original skewed training data as a hold out set, balanced the rest and trained on it, and calculated the AUC on the hold out set. We repeated this procedure n times ($n \approx 100$) and found the population mean μ and standard deviation σ. We then modeled the mean of our AUC scores as a Gaussian with standard deviation $\frac{\sigma}{\sqrt{n}}$, which yields the 95% confidence interval $\mu \pm 1.96\frac{\sigma}{\sqrt{n}}$.

V. Results

A. FINDRISC and Logistic Regression

FINDRISC is a logistic regression model on seven features: age, BMI, waist circumference, use of antihypertensive agents, history of hyperglycemia, physical activity, and dietary patterns [10]. Only age, BMI, and the use of hypertensive agents were present in our data set. We ran logistic regression using only these three features to get the performance of FINDRISC on our data set. We also ran logistic regression on our full feature set, excluding binary features where one category only appeared a few times. The results can be found in Table I.

TABLE I: Logistic Regression

Feature Set	CV AUC	95% CI	Test AUC
FINDRISC	0.7566	$\pm 3.578 \times 10^{-3}$	0.7621
Full	0.8056	$\pm 2.778 \times 10^{-3}$	0.8140

We note that in the original FINDRISC study where patients were screened to collect all required data, the AUC was 0.87 [10].

B. Naive Bayes

We used the e1071 R package for Naive Bayes classification. The package models continuous features as Gaussians with a different mean and variance for each class to be predicted. We used Laplacian smoothing with a smoothing parameter of 1.

C. k-Nearest Neighbor

We used Matlab's ClassificationKNN library to apply the k-NN algorithm with different distance functions. For each distance function, we tried all odd k between 1 and $\sqrt{\text{number of training samples}} \approx 101$, as suggested by literature [11]. The AUC values of the different distance formulas can be seen in Fig. 1.

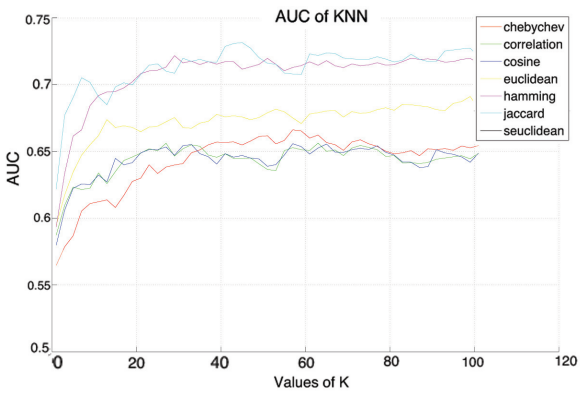

FIG. 1: AUC as a function of k. Each plot represents a different distance function for KNN.

D. Random Forest

We used the implementation of Breiman's Random Forest algorithm in the RandomForest R package. Using grid search with 10-fold cross-validation, we chose the number of trees in the forest and the number of features to randomly select at each node as candidates for splitting.

E. GBM

We fit a gradient boosting machine (GBM) model using the gbm R package. We estimated the optimal number of trees through cross-validation using the dismo package.

F. SVM

1. Choice of Primal Problem

Research suggested that for skewed data we should use different regularization hyper parameters for positive and negative classes [12]. We followed this approach instead of balancing the training data through duplication. We thus solved the following two-class primal problem, using LIBSVM:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}w^T w + C_1 \sum_{y_i=1} \xi_i + C_2 \sum_{y_i=-1} \xi_i \\ \text{subject to} \quad & y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, i = 1, \ldots, m \end{aligned}$$

We used the Radial Basis Function (RBF) as a kernel, for reasons discussed in *2*:

$$K(x_i, x_j) = e^{-\gamma ||x_i - x_j||^2}$$

We assigned the hyper parameter C_1 as the weight for diabetic patients and C_2 as the weight for non-diabetic patients. We experimented with different weights by doing a grid search for the best value for C_1 and γ, keeping C_2 pegged to C_1. The 20-fold cross-validation AUC can be seen in Table II. Note that these calculations were done on a limited feature set composed of only the ICD9 and NDC codes for the sake of time.

TABLE II: Best AUC for Weight Ratios - $\frac{C_1}{C_2}$

Ratio - $\frac{C_1}{C_2}$	0.5	2	4	8	32	128
Best CV AUC	0.783	0.803	0.806	0.802	0.772	0.739

The AUC value peaks for the ratio $\frac{C_1}{C_2} = 4$. We noted that this is almost the ratio of non-diabetic patients to diabetic patients in our data set. This makes sense, as by setting the regularization parameter for diabetic patients 4 times that of non-diabetic patients, we are effectively counting diabetic patients 4 times in the objective function. Thus, it has the same effect as training with a single-class primal problem, but on a modified dataset where each diabetic patient record is replicated 3 times. For all future runs, the weight ratio $\frac{C_1}{C_2}$ was set to 4.

2. Choice of Kernel

Research suggested using the RBF kernel as a default first choice, given that none of the following conditions hold true [14]:

1. The number of instances is much smaller than the number of features.
2. Both the number of instances and features are large (both in 10^5).
3. The number of instances is much greater than the number of features.

With 811 features and 7958 training instances, using the RBF was justified. We nonetheless verified that RBF was superior to other common kernels. Table III shows the best 20-fold CV AUC for each kernel.

TABLE III: Best AUC for Kernels

Kernel	Linear	Polynomial	RBF	Sigmoid
Best CV AUC	0.502	0.742	0.817	0.812

3. Implementation Challenges

When we first ran our SVM on the test set, the test AUC was much lower than the CV AUC. To investigate whether the difference was due to the particular way our data was divided into training and test sets, we created new test and training sets from the training data and repeated the process of selecting hyper parameters and evaluating the final SVM. Again the test AUC was much lower than the CV AUC.

We ultimately fixed the problem by using 20-fold cross-validation instead of 5-fold. This solution worked because the best choice of hyper parameters depends on the size of the training set [13]. The hyper parameters we found to be the best for 80% of the training set were not the best for the full training set. Ideally, to fix the problem we would use leave-one-out cross-validation, but that was not computationally feasible.

G. Ensemble Methods

We constructed ensemble models by taking the mean and median of the predictions from the top k models for each patient. We excluded SVM for lack of time. We found that the mean of the top two models, i.e. random forest and GBM, performed best. Including more models decreased performance, though the difference between $k = 2$ and $k = 3$ was not statistically significant.

TABLE IV: Ensemble CV AUC with 95% CI for different methods and k

	k = 2	k = 3	k = 4
Mean	0.845 ± 0.005	0.839 ± 0.005	0.830 ± 0.005
Median	0.845 ± 0.005	0.841 ± 0.004	0.834 ± 0.005

H. Summary

TABLE V: Model Performance on Full Feature Set

Model	CV AUC	95% CI	Test AUC
FINDRISC	0.757	$\pm 3.58 \times 10^{-3}$	0.762
Logistic Regression	0.806	$\pm 2.78 \times 10^{-3}$	0.814
Naive Bayes	0.793	$\pm 3.62 \times 10^{-3}$	0.805
k-Nearest Neighbor	0.701	$\pm 5.20 \times 10^{-5}$	0.732
Random Forest	0.836	$\pm 3.18 \times 10^{-3}$	0.839
GBM	0.831	$\pm 3.16 \times 10^{-3}$	0.846
RBF Kernel SVM	0.817	$\pm 3.89 \times 10^{-3}$	0.816
Ensemble	0.845	$\pm 4.81 \times 10^{-3}$	0.849

The AUC results for all models discussed in this paper can be seen in Table V. The ensemble of random forest and GBM performed best on both the training and test sets.

TABLE VI: Model Performance on Top 10 Features

Model	CV AUC	95% CI	Test AUC
Logistic Regression	0.802	$\pm 3.13 \times 10^{-3}$	0.814
Naive Bayes	0.799	$\pm 3.46 \times 10^{-3}$	0.807
k-Nearest Neighbor	0.715	$\pm 5.40 \times 10^{-3}$	0.733
Random Forest	0.790	$\pm 2.97 \times 10^{-3}$	0.796
GBM	0.805	$\pm 3.79 \times 10^{-3}$	0.817
RBF Kernel SVM	0.784	$\pm 3.95 \times 10^{-3}$	0.761

Table VI shows the models' results on the top 10 features. The logistic regression and GBM models performed best on both the test and training sets. The difference between the two models' cross-validation AUCs was not statistically significant.

Note that the test AUC was outside the 95% confidence interval for many models. This happened because the CI was for the models' AUC scores on data from the training set, which is a slightly different distribution than AUC scores on a random data set, such as the test set. Thus neither CV nor test AUC give us a perfect measure of model quality.

VI. Conclusion

A. Best Model

Our most successful model was the ensemble of random forest and GBM trained on the full feature set. Our ensemble model achieved an AUC of 0.849 on the test set, which vastly outperforms logistic regression with the FINDRISC features. Our model is competitive with FINDRISC screening in the field, where studies have found an AUC of 0.87 [10]. This shows that general EHR are sufficient to detect undiagnosed diabetes. Automated diabetes detection can not only complement but even compete with more expensive screening programs.

In order to use our model to identify undiagnosed diabetes, we must select a threshold probability at which to classify a patient as diabetic. The optimal threshold depends on how we value precision relative to recall. A medical provider could determine the threshold by comparing the cost of testing a non-diabetic patient for diabetes with the cost of failing to diagnose a diabetic. Table VII shows the thresholds that maximize the F_β score, where higher β favors recall over precision. Fig. 2 shows the model's performance over all thresholds.

TABLE VII: Optimal Thresholds

Criteria	Score	Threshold	Precision	Recall
F_1	0.584	0.429	0.489	0.725
F_2	0.697	0.225	0.355	0.919

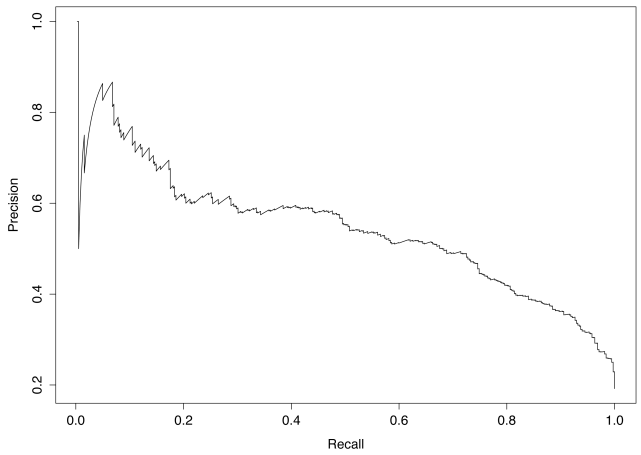

FIG. 2: Precision vs. recall as the classification threshold decreases.

B. Comparing Models and Feature Sets

The complex models, i.e. random forest, SVM and GBM, performed better on the full feature set than on the top 10 features. By contrast, the simple models, i.e. logistic regression, naive Bayes, and k-nearest neighbor, performed equally on both feature sets. These results accord with the fact that the simpler models have higher bias than the more complex models. Random forest and SVM performed worse than the simpler models on the top 10 features, which suggests that the models overfit the data due to their higher variance.

Logistic regression on the top 10 features did not perform as well as FINDRISC in the field. Thus we would not recommend using our top 10 features as a screen-

ing heuristic. If medical providers' EHR lack important predictors of diabetes, as our data set did, they will get better results by generating a large feature set and fitting complex models.

VII. Future Work

We could improve our results by generating more features. All our best models were around 0.84 CV AUC, which suggests that we hit the limit on how well we can predict diabetes with our feature set. We suspected that our feature set was our main limitation after developing random forest and SVM models. But we continued to apply new models rather than mine for more features because those results were more interesting and could be generalized to other data sets.

By contrast, any further features we created would have been highly specific to our data set. For example, Kaggle removed blood glucose lab tests from the data set because they reveal whether patients have diabetes. We could thus have searched for features that indicate patients had lab tests removed, since people who take a blood glucose test are more likely to have diabetes.

Collecting more data would also improve our models. Fig. 3 shows a learning curve for random forest and GBM. At 100% of the training set, the slope of the plotted test AUC is small but positive for both models. We did not include training AUC for random forest because the model predicted the training set perfectly.

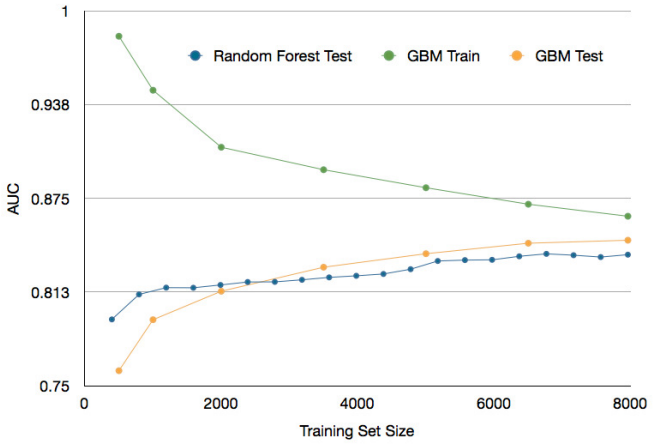

FIG. 3: Learning curve for random forest and GBM.

Further exploration of ensemble methods could yield better models. We could construct many random forest and GBM models with different hyper parameters or trained on different subsets of the training data and combine them in more sophisticated ways. In addition, we could improve our GBM models by investigating ways of customizing them besides changing the number of trees.

The next step for our work is to try our approach in the field. We could collaborate with a medical provider to build a predictive model from their EHR. This would teach us how well our methods generalize across data sets. Further, we could experiment with different ways of assembling a training set. For instance, could we get better results if we only trained on patients who were tested for diabetes? Our data set had patients who were not tested, and some of them may have had undiagnosed diabetes. We could also tackle the problem of estimating how likely patients are to develop diabetes.

Acknowledgements

We acknowledge the winners of the "Practice Fusion Diabetes Classification" challenge for inspiration for features and models [6]. We thank Andrew Ng, Andrew Maas, all CS 229 TAs, and Dr. Ajay Sood for their advice.

References

[1] "Diabetes." WHO. N.p., n.d. Web. 13 Dec. 2012. <http://www.who.int/mediacentre/factsheets/fs312/en/>.

[2] Centers for Disease Control and Prevention. Centers for Disease Control and Prevention, n.d. Web. 13 Dec. 2012. <http://www.cdc.gov/media/pressrel/2010/r101022.html>.

[3] World Health Organization. Guidelines for the prevention, management and care of diabetes mellitus. In: Khatib OM, editor. Vol. 32. EMRO Technical Publications Series; 2006.

[4] "ADA: Diabetes Screening Cost Effective." Diabetes In Control. N.p., n.d. Web. 13 Dec. 2012. <http://www.diabetesincontrol.com/articles/diabetes-news/9532-ada-diabetes-screening-cost-effective>.

[5] Schwarz, P., J. Li, J. Lindstrom, and J. Tuomilehto. "Tools for Predicting the Risk of Type 2 Diabetes in Daily Practice." Hormone and Metabolic Research 41.02 (2009): 86-97. Print.

[6] "Practice Fusion Diabetes Classification." Data. N.p., n.d. Web. 13 Dec. 2012.

[7] "International Classification of Diseases, Ninth Revision." Centers for Disease Control and Prevention. Centers for Disease Control and Prevention, 01 Sept. 2009. Web. 13 Dec. 2012. <http://www.cdc.gov/nchs/icd/icd9.htm>.

[8] "Clinical Classifications Software (CCS) for ICD-9-CM." HCUP-US Tools & Software Page. N.p., n.d. Web. 13 Dec. 2012. <http://www.hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp>.

[9] "National Drug Code Directory." FDA. N.p., n.d. Web. 13 Dec. 2012. <http://www.fda.gov/drugs/informationondrugs/ucm142438.htm>.

[10] Herman WH, Smith PJ, Thompson TJ, Engelgau MM, Aubert RE. A new and simple questionnaire to identify people at increased risk for undiagnosed diabetes. Diabetes Care 1995; 18: 382-387

[11] Qi H., Feature selection and kNN fusion in molecular classification of multiple tumor types, International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences, June 2002.

[12] Osuna, R. Freund, and F. Girosi. Support vector machines: Training and applications. AI Memo 1602, Massachusetts Institute of Technology, 1997.

[13] Maas, Andrew. Office Hours, 229.

[14] Hsu, Chih-Wei, Chih-Chung Chang, and Chih-Jen Lin. "A Practical Guide to Support Vector Classification." N.p., n.d. Web. 13 Dec. 2012. <http://www.csie.ntu.edu.tw/ cjlin/papers/guide/guide.pdf>.