Predicting Disease Risks from Highly Unbalanced Data using Random Forest

Mohammed Khalilia¹, Sounak Chakraborty², Mihail Popescu³

¹Computer Science Department, University of Missouri, Columbia, Missouri, USA
²Statistics Department, University of Missouri, Columbia, Missouri, USA
³Health Informatics Department, University of Missouri, Columbia, Missouri, USA

Email addresses:
MK: makkx9@mail.missouri.edu
CH: chakrabortys@missouri.edu
MP: popescum@missouri.edu
Abstract

Background
We present a method using random forest (RF) for predicting disease risk of individuals based on their medical history. This method can be incorporated in a variety of applications such as risk management, tailored health communication and decision support systems in healthcare.

Methods
We used medicare data, which is publicly available through Healthcare Cost and Utilization Project (HCUP). The data set is highly unbalanced. Therefore, in order to overcome the class imbalance problem, we used an ensemble learning method that consist in repeated random sub-sampling. This technique divides the training data into multiple sub-samples, allowing each each sub-sample to be fully balanced. We compared the performance of support vector machine (SVM) and RF to predict the risk of eight chronic diseases.

Results
We performed prediction on eight disease categories. The RF ensemble learning method outperformed on average SVM by 5.5% in terms of sensitivity and by 1.8% in terms of the area under the ROC curve (AUC). In addition, RF has the advantage of computing the importance of each variable in the classification process.

Conclusions
In combining repeated random sub-sampling with RF, we were able to overcome the class imbalance problem and achieve promising results. Using the HCUP data set, we predicted eight disease categories with an average AUC of 88.58%.
**Background**

Disease prediction can be applied to different domains such as risk management, tailored health communication and decision support systems. Risk management plays a role in health insurance companies, mainly in the underwriting process [1]. Health insurers use a process called underwriting in order to classify the applicant as standard or substandard, based on which they compute the policy rate and the premiums individuals have to pay. Currently, in order to classify the applicants, insurers require every applicant to complete a questionnaire, report current medical status and sometimes medical records, or clinical laboratory results, such as blood test, etc. Incorporating machine learning techniques, insurers can make evidence based decisions and can optimize, validate and refine the rules that govern their business. For instance, Yi et al. [2], applied association rules and SVM on a insurance company database to classify the applicants as standard, substandard or declined.

Another important aspect where disease prediction can be applied is tailored health communication. For example, one can target tailored educational materials and news to a subgroup, within the general population, that has specific disease risks. Cohen et al. [3], discussed how tailored health communication for cancer patients can motivate cancer prevention and early detection. Disease risk prediction along with tailored health communication can lead to an effective channel for delivering disease specific information for people who will be likely to need it.

Several machine learning techniques were applied to health care data sets for the prediction of future health care utilization such as predicting individual expenditures and disease risks for patients. Moturu et al. [4], predicted future high-cost patients based on data from Arizona Medicaid program. They used random data sampling to
overcome the problem of unbalanced data along with variety of classification methods such as SVM, Logistic Regression, Logistic Model Trees, AdaBoost and LogitBoost. Davis et al. [5], used clustering and collaborative filtering to predict individual disease risks based on medical history. The prediction is performed multiple times for each patient with each iteration employing different basis for clustering; then the clusterings are combined to form an ensemble. The output is a ranked list of diseases that can be used in subsequent visits of that patient. Mantzaris et al. [6], predicted Osteoporosis using Artificial Neural Network (ANN). They used two different ANN techniques: Multi-Layer Perceptron (MLP) and Probabilistic Neural Network (PNN). Hebert et al. [7], identified persons with diabetes using Medicare claims data. They ran into a problem where the diabetes claims occur too infrequently to be sensitive indicators for persons with diabetes. In order to increase the sensitivity, physician claims where included. Yu et al. [8], illustrates a method using SVM for detecting persons with diabetes and pre-diabetes.

Zhang et al. [9], conducted a comparative study of ensemble learning approaches. They compared AdaBoost, LogitBoost and RF to logistic regression and SVM in the classification of breast cancer metastasis. They concluded that ensemble learners have higher accuracy compared to the non-ensemble learners.

Together with methods for predicting disease risks, in this paper we discuss a method for dealing with highly unbalanced data. Earlier, we mentioned two examples [4, 7] where the authors encountered class imbalanced problems. Class imbalance occurs if one class contains significantly more samples than the other class. Since classifiers assume that the data is drawn from the same distribution as the training data which is
not the case in class-imbalanced data sets, they will produce undesirable results in such situations. The Medicare data set we use in this paper is highly unbalanced; the majority of the patients do not have a disease, while a very small subset does. For example, only 3.59% of the patients have heart disease, thus it is possible that a classifier learning from this data might achieve an accuracy of 96.41% while the sensitivity of 0%.

Data
The Nationwide Inpatient Sample (NIS) is a database of hospital inpatient admissions dated back to 1988 and is used to identify, track, and analyze national trends in health care utilization, access, charges, quality, and outcomes. The NIS database is developed by the Healthcare Cost and Utilization Project (HCUP) and sponsored by the Agency for Healthcare Research and Quality (AHRQ) [10]. This database is publicly available and does not contain any patient identifiers. The database contains discharge level information on all inpatients from a 20% stratified sample of hospitals across the United States, representing approximately 90% of all hospitals in the country [10]. The five strata for hospitals are based on the American Hospital Association classification. HCUP data from the year 2005 will be used in this paper.

The data set contains 7,995,048 hospital stays and 126 clinical and nonclinical data elements for each hospital stay. Nonclinical elements include patient demographics, hospital identification, admission date, zip code, calendar year, total charges and length of stay. Clinical elements include procedures, procedure categories, diagnosis codes and diagnosis categories. Every record contains a vector of 15 diagnosis codes. The diagnosis codes are represented using the International Classification of Diseases, Ninth Revision, Clinical Modification (ICD-9-CM). The International
Statistical Classification of Disease is designed and published by the World Health Organization (WHO). The ICD-9 codes are alphanumeric codes, 3-5 characters long and used by hospitals, insurance companies and other facilities to describe health conditions of the patient. Every code represents a disease, condition, symptom, or cause of death.

In addition, every record contains a vector of 15 diagnosis category codes. The diagnosis categories are based on Clinical Classification Software (CCS) developed by HCUP in order to categorize the ICD-9 diagnosis and procedures. There are numerous codes, over 14,000 ICD-9 codes and 3,900 procedures codes; CCS collapsed these codes into a smaller number of clinically meaningful categories, called diagnosis categories. Every ICD-9 code has a corresponding diagnosis category and every category contains a set of ICD-9 codes. There are 259 diagnosis categories, every category is denoted by a value in the range 1-259. Demographics such as age, race and sex are also included in the data set. Figure 1 shows the distribution of patients across age, race and sex. These three demographic variables are important in predicting certain medical conditions.

The data set is highly unbalanced. The unbalance rate ranges from 0.01-29.1%. 222 diagnosis categories occurs in less than 5% of the patients and only 13 categories
occur in more than 10% of the patients. Table 1 shows the top 10 most prevalent disease categories and Table 2 shows some of the rarest diseases in the 2005 HCUP data set.

Table 1 - The 10 most prevalent diseases

| Disease                                | Prevalence |
|----------------------------------------|------------|
| Hypertension                           | 29.1%      |
| Coronary Atherosclerosis                | 27.65%     |
| Hyperlipidemia                         | 14.46%     |
| Dysrhythm                              | 14.35%     |
| Other Circulatory Diseases             | 12.02%     |
| Diabetes mellitus no complication      | 12%        |
| Anemia                                 | 11.93%     |

Table 2 - Some of the most unbalanced diseases

| Disease                                | Percent of Active class |
|----------------------------------------|-------------------------|
| Male Genital Disease                   | 0.01%                   |
| Testis Cancer                          | 0.046%                  |
| Encephalitis                           | 0.059%                  |
| Aneurysm                               | 0.74%                   |
| Breast Cancer                          | 1.66%                   |
| Peripheral Atherosclerosis             | 3.16%                   |
| Diabetes Mellitus w/complication       | 4.7%                    |
| Other Circulatory Diseases             | 5.6%                    |

Some limitations of the 2005 HCUP data set is the arbitrary order in which the ICD-9 codes and disease categories were listed. The codes were not listed in the chronological order according to the date they were diagnosed. Also, the data set does not provide anonymous patient identifier, which could be used to check if multiple records belong to the same patient or to determine the elapsed time between diagnoses.

**Methods**

**Data Pre-processing**

The data set was provided in a large ASCII file containing the 7,995,048 records. The first step was to parse the data set, randomly select $N$ records and extract a set of
relevant features. Every record is a sequence of characters that are not delimited. However, the data set instructions specifies the starting column and the ending column in the ASCII file for each data element (length of data element). HCUP provides a SAS program to parse the data set, but we chose to develop our own C program to perform the parsing.

**Feature Selection**
For every record, we extracted the age, race, sex and 15 diagnosis categories. As mentioned before, there are 259 categories, every record will be transformed to \( m=262 \) dimensional feature vector. Features 1-259 are binary and each represents a disease category. A patient either has the disease (1) (belongs to the "active" class) or does not have the disease (0). The other three features are age, race and sex.

**Learning from Unbalanced Data**
A data set is class-imbalanced if one class contains many more samples than the other class. As mentioned above the data set is highly unbalanced (Table 2). The unbalance rate of the data ranges between 0.01-29.1%, where this percentage represents the percent of the data samples that belong to the active class. For example, 3.16% of the patients have Peripheral Atherosclerosis while the rest do not. In such cases, it will be hard create an appropriate testing and training data sets given that most classifiers are built with the assumption that the test data is drawn from the same distribution as the training data [11].

Presenting unbalanced data to the classifier will produce undesirable results. Most likely that learning from unbalanced data will cause the classifier to have a much lower performance on the testing that on the training data. There exist techniques to learn classifiers for unbalanced data such as oversampling, undersampling, boosting,
bagging and repeated random sub-sampling. We will briefly describe each of these techniques, but the focus will be on the repeated random sub-sampling.

**Oversampling and Undersampling**
In oversampling the number of examples from the small class is increased to reach the size of the larger class, while in undersampling the examples from the large class are reduced to reach the size of the smaller class. In this paradigm, one need to make a decision whether to use oversampling, undersampling, or a combination of both, and select the appropriate rate to perform the sampling. For example, we could perform oversampling and undersampling to full balance (equal number of samples from each class) or we could choose a different ratio of the two classes [12].

**Bagging**
Bagging constructs a number, $T$, of weak classifiers [13]. For each weak classifier $t=1,2,...,T$ a training data of size $N$ is sampled from the original data set with replacement. The size of the sampled data is the same size as the original training data, some samples may appear more than once, other may not appear at all. The $T$ classifiers are then aggregated to form a stronger classifier.

**Boosting**
Unlike bagging where independent bootstrap samples are selected from the original training data, boosting associates a weight with every instance [13]. The classifier will use the weights to focus on the samples that are most difficult to classify. For every weak classifier $t=1,2,...,T$, the weights are updated, that is the weights of misclassified instances are increased. At the end, the weak classifiers are aggregated to form a strong classifier using weighted voting.

**Repeated Random Sub-Sampling**
Repeated random sub-sampling was found to be very effective in dealing with data sets that are highly unbalanced. Because most classification algorithms make the assumption that the class distribution in the data set is uniform, it is essential to pay
attention to the class distribution when addressing medical data. This method divides the data set into training and testing data, then it partitions the training data into sub-samples with each sub-sample containing an equal number of instances from each class, except for last sub-sample, in some cases. The model is fitted repeatedly on every sub-sample and the final result is a majority voting over all the sub-samples. In this paper we used repeated random sub-sampling approach to randomly choose $N$ samples from the original HCUP data set. The testing data contains 30% of $N$ and the training data contains the remaining 70%. The $N$ samples are divided into two separate data sets, $N_1$ active data samples and $N_0$ inactive data samples, where $N_1 + N_0 = N$. The testing data will contain 30% of the $N_1$ ($TsN_1$) active samples while the remaining 70% will be sampled from the $N_0$ ($TsN_0$) inactive samples. The training data set will contain the remaining active samples ($TrN_1$) and inactive samples ($TrN_0$).

Since the training data is highly unbalanced ($TrN_1 << TrN_0$), the $TrN_0$ samples are partitioned into $NoS$ training sub-samples, where $NoS$ is the ratio between $TrN_0$ and $TrN_1$. Every sub-sample has equal number of instances of each class. The training active samples ($TrN_1$) are fixed among all the training data sub-samples, while the inactive samples will be sampled without replacement from $TrN_0$. There will $NoS$ sub-samples to train the model on. Eventually, every inactive sample in the training data is selected once, while every active sample is selected $NoS$ times. After training the model on all the sub-samples, we take the majority voting to determine the final votes (Algorithm 1).
Algorithm 1 Repeated Random Sub-Sampling

1: \( TsN = \text{total number of samples in the testing data} \)
2: \( N_0 = \text{number of inactive samples} \)
3: \( N_1 = \text{number of active samples} \)
4: \( N = \text{total number of samples in the data set, where } N=N_0+N_1 \)
5: Generate testing data
   a. Randomly select \( TsN_1 \) samples from \( N_1 \), where \( TsN_1=0.3*N_1 \)
   b. Randomly select \( TsN_0 \) samples from \( N_0 \), where \( TsN_0=TsN - TsN_1 \)
6: \( Ts = TsN_0 \) samples + \( TsN_1 \) samples
7: Generate training data
   a. Contains \( TrN_1 \) samples, \( TrN_1 = \text{remaining } N_1 \) samples after generating testing data
   b. Contains \( TrN_0 \) samples, \( TrN_0 = \text{remaining } N_0 \) samples after generating testing data
8: \( NoS = TrN_0/TrN_1 \)
9: for \( s = 1 \) to \( NoS \) do
   a. Generate the \( s \) training data sub-sample
      i. \( TrSS_1 = \text{All } TrN_1 \) samples (Active samples)
      ii. \( TrSS_0 = \text{Randomly select } TrN_1 \) samples from \( TrN_0 \) samples without replacement (Guarantees full balance of the training data sub-samples, except for the last sub-samples, in some cases)
   b. \( TrSS = TrSS_0 + TrSS_1 \)
   c. \( y_s(x) = \text{classifier}(TrSS, Ts) \) (Predicted class labels for \( Ts \) using sub-sample \( TrSS \))
10: end for
11: \( y(x) = \text{majority voting } \{y_s(x)\}_{s=1}^{NoS} \) (Final predicted class is majority voting over all sub-samples)

Random Forest
RF is an ensemble learner, a method that generates many classifiers and aggregates their results. RF adds a layer of randomness to bagging by building large collection of de-correlated trees. Due to this randomness, RF is usually faster than bagging. RF will create multiple CART-like trees, each trained on a bootstrap sample of the original training data and searches across a randomly selected subset of input variables to determine the split. Each tree in RF will cast a vote for some input \( x \), then the output of the classifier is determined by majority voting of the trees (Algorithm 2). RF can handle high dimensional data and use a large number of trees in the ensemble. Some important features of RF as mentioned in [14]:

1. It has an effective method for estimating missing data.
2. It has a method for balancing error in unbalanced data.
3. It estimates the importance of variables used in the classification.

Algorithm 2 Random Forest for Classification

1: \( n_{\text{tree}} = \) number of trees to be generated
2: \( N = \) number of samples in the data set
3: \begin{align*}
\text{for } t = 1 \text{ to } n_{\text{tree}} \text{ do} \\
& \text{a. Generate bootstrap sample } Z \text{ of size } N \text{ from the original data - with replacement} \\
& \text{b. for each bootstrap } Z \text{ grow a classification tree} \\
& \text{c. for } i = 1 \text{ to } \text{NumberOfNodes} \text{ do} \\
& \quad \text{i. randomly sample } m_{\text{try}} \text{ variables from } M \text{ variables} \\
& \quad \text{ii. choose best split among the sampled variables (bagging is special case of RF and obtained when } m_{\text{try}} = M) \\
& \text{d. end for} \\
& \text{e. } y_t(x) = \text{class prediction of the } t^{th} \text{ tree} \\
\text{end for} \\
4: \ Y_{\text{RF}}(x) = \text{majority voting } \{y_t(x)\}_{t=1}^{n_{\text{tree}}} \text{ (Final predicted class is majority voting over all trees in RF)} \\

We attempted to use weighted random forest (WRF), but the results were not satisfactory. That could be due to poor selection of weights since weights in WRF are tuning parameters. Chen et al. [15], compared WRF and balanced random forest (BRF) on six different and highly unbalanced data sets. In WRF, they tuned the weights for every data set, while in BRF, they changed the votes cutoff for the final prediction. They concluded that BRF is computationally more efficient than WRF for unbalanced data. They also found that WRF is more vulnerable to noise compared to BRF. In this paper, we used RF without tuning the class weights or the cutoff parameter. In order to address the class imbalanced problem we used repeated random sampling, as discussed in the previous section.

Splitting Criterion

Like CART, RF uses the Gini measure of impurity to select the split with the lowest impurity at every node [16]. The Gini measure of impurity measures the impurity of a variable with respect to the classes, the measure produces a value in the range \([0, 1]\). Formally, the Gini impurity measure for variable \( X = \{x_1, x_2, \ldots, x_j\} \) at node \( t \), where \( j \) is the number of children at node \( t \), \( N \) is the number of samples, \( n_{ci} \) is the number of
samples with value $x_i$ belonging to class $c$, $a_i$ is the number of samples with value $i$ at node $t$, then the Gini index is given by [17]

$$I(t_{x_i}) = 1 - \sum_{c=0}^{c} \left( \frac{n_{ci}}{a_i} \right)^2$$

(1)

The Gini index of a split is the weighted average of the Gini measure over the different values of variable $X$, which is given by

$$Gini(t, X) = \sum_{i=1}^{j} \frac{a_i}{N} I(t_{x_i})$$

(2)

The decision of the splitting criterion will be based on the lowest Gini impurity value computed among all the variables.

**Variable Importance**

One of the most important features of RF is the output of the variable importance. Variable importance measures the degree of association between the variable $m$ and the classification result. RF has four measures for the variable importance: raw importance score for class 0, raw importance score for class 1, decrease in accuracy and the Gini index. To estimate variable importance for some variable $m$, the out-of-bag (OOB) samples are passed down the tree and the prediction accuracy is recorded. Then the values for variable $m$ are permuted in the OOB samples and the accuracy is measured again. These calculations are carried out tree by tree as the RF is constructed. The average decrease in accuracy of these permutations is then averaged over all the trees and is used to measure the importance of the variable $m$. If the prediction accuracy decreases substantially, then that suggests that the variable $m$ has strong association with the response [18]. After measuring the importance of all the variables, RF will return a ranked list of the variable importance.
Formally, let $\beta_t$ be the OOB samples for tree $t$, $t \in \{1, \ldots, n_{tree}\}$, $y_i'$ is the predicted class for instance $i$ before the permutation in tree $t$ and $y_i'^{t,a}$ is the predicted class for instance $i$ after the permutation. The variable importance $VI$ for variable $m$ in tree $t$ is given by

$$VI_m^t = \frac{\sum_{i=1}^{N} \beta_t I(y_i = y_i'^t) - \sum_{i=1}^{N} \beta_t I(y_i = y_i'^{t,a})}{|\beta_t|}$$

The raw importance value for variable $m$ is then averaged over all trees in the RF.

$$VI_m = \frac{\sum_{t=1}^{n_{tree}} VI_m^t}{n_{tree}}$$

The variable importance used in this paper is the Mean Decrease Gini (MDG), which is based on the Gini splitting criterion discussed earlier. The MDG measure the decrease $\Delta I$ (equation 1) that results from the splitting. For two class problem, the change in $I$ at node $t$ is defined as [19]

$$\Delta I(t) = I(t_{x_i}) - Gini(t,X)$$

Where $I(t_{x_i})$ is defined in equation 1 and $Gini(t,X)$ is defined in equation 2. The decrease in Gini impurity is recorded for all the nodes $t$ in all the trees ($n_{tree}$) in RF for all the variables and Gini Importance ($GI$) is then computed as [19]

$$GI = \sum_{n_{tree}} \sum_{t} \Delta I(t)$$

**Classification with Repeated Random Sub-Sampling**

Training the classifier on a data set that is small and highly unbalanced will result in unpredictable results as discussed in earlier sections. To overcome this issue, we used repeated random sub-sampling. Initially, we construct the testing data and the NoS
training data sub-samples. The classifier will be learned $NoS$ number of times, in every iteration the input to the classifier is one of the training data sub-samples and the testing data. Each iteration gives a classification and final vote is computed using majority voting over all the splits.

**Model Evaluation**
To evaluate the performance of the RF we compared it to SVM on unbalanced data sets for eight different chronic diseases categories. Two sets of experiments were carried out:

1. Set I: this scheme compares RF and SVM performance with repeated random sub-sampling. Both classifiers were fitted to the same training and testing data and the process was repeated 100 times. The ROC curve and the average AUC for each classifier were calculated and compared.

2. Set II: this scheme compares RF performance with and without the sampling approach. Without sampling the data set is highly unbalanced, while sampling should improve the accuracy since the training data sub-samples fitted to the model are balanced. The process was again repeated 100 times and the ROC curve and the average AUC were calculated and compared.

**Results**
We performed the classification using R, which is an open source statistical software. We used R randomForest and SVM (e1071) packages. The parameters to the RF were as follows: number of trees ($ntree$) was set to 500. Overall, the number of trees did not highly influence the classification results. We verified that by running RF for different $ntree$ values to predict breast cancer, we plotted $ntree$ versus sensitivity (Figure 2). The number of variables randomly sampled as candidates at each split ($mtry$) is the square root of the number of features, in our case for 262 features, $mtry$ was set to 16.
Palmer et al. [20] and Liaw et al. [21] also reported that RF is usually insensitive to the training parameters. Figure 2 shows how sensitivity in RF varies as the number of trees ($ntree$) varies, we varied $ntree$ from 1-1001 in intervals of 25 and measured the sensitivity at every interval. Sensitivity ranged from 0.8457 when $ntree = 1$ and 0.8984 when $ntree = 726$. In our experiments we used $ntree = 500$ since the $ntree$ did not have a large effect on accuracy for $ntree > 1$.

![Figure 2 - RF behaviour when the number of trees ($ntree$) varies](image.png)

For SVM we used a linear kernel, termination criterion ($tolerance$) was set to 0.001, $epsilon$ for the insensitive-loss function was 0.1 and the regularization term ($cost$) was 1.

We randomly selected $N=10,000$ data points from the original HCUP data set. We predicted the disease risks on 8 out of the 259 disease categories. Those categories are: breast cancer, type 1 diabetes, type 2 diabetes, hypertension, coronary atherosclerosis, peripheral atherosclerosis, other circulatory diseases and osteoporosis.
Result set I: Comparison of RF and SVM

Both SVM and RF classification were performed 100 times and the average area under the curve (AUC) was measured. The repeated random sub-sampling approach has improved the detection rate considerably. On seven out of eight disease categories, RF outperformed SVM in terms of AUC (Table 3). In addition, RF has ROC curves that reach 100% detection rate faster than SVM (Figures 3, 4, 5), that is we obtain 100% detection rate at a lower false alarm rate.

Table 3 - RF vs. SVM performance in terms of AUC on eight disease categories

| Disease                          | RF     | SVM    |
|---------------------------------|--------|--------|
| Breast cancer                   | 0.9017 | 0.8976 |
| Diabetes no complication        | 0.868  | 0.8355 |
| Diabetes with/complication      | 0.9335 | 0.9205 |
| Hypertension                    | 0.8975 | 0.8465 |
| Coronary Atherosclerosis        | 0.923  | 0.8933 |
| Peripheral Atherosclerosis      | 0.9069 | 0.8948 |
| Other Circulatory Diseases      | 0.7846 | 0.7724 |
| Osteoporosis                    | 0.8708 | 0.8805 |
Figure 3 - ROC curve for diabetes mellitus

ROC curve for diabetes mellitus comparing both SVM and RF, average AUC over the 100 runs is 87.8% and 90% respectively.
Figure 4 - ROC curve for hypertension

ROC curve for hypertension comparing both SVM and RF, average AUC over the 100 runs is 84.65% and 89.75% respectively.
Figure 5 - ROC curve for breast cancer

ROC curve for breast cancer comparing both SVM and RF, average AUC over the 100 runs is 89.76% and 90.17% respectively.

We compared our results to the ones reported by other authors. For instance, Yu et al. [8], illustrates a method using SVM for detecting persons with diabetes and prediabetes. They used data set from the National Health and Nutrition Examination Survey (NHANES). NHANES collects demographic, health history, and behavioural information, it may also include detailed physical, physiological, and laboratory examinations. The AUC for their classification scheme I and II was 83.47% and 73.81% respectively. We also predicted diabetes using SVM and RF and the AUC values were 87.8% and 90% respectively (ROC curve in Figure 3).
Davis et al. [5] used clustering and collaborative filtering to predict disease risks of patients based on their medical history. Their algorithm generates a ranked list of diseases in the subsequent visits of that patient. They used an HCUP data set, similar to the data set we used. Their system predicts more than 41% of all the future diseases in the top 20 ranks. One reason for their low system performance might be that they predict the exact ICD-9 code for the patient, while we predict the disease category.

Zhang et al. [9] performed classification on breast cancer metastasis. In their study, they used two published gene expression profiles. They compared multiple methods (logistic regression, SVM, AdaBoost, LogitBoost and RF). In the first data set, the AUC for SVM and RF was 88.6% and 89.9% respectively and for the second data set 87.4% and 93.2%. We also performed prediction on breast cancer, the AUC for SVM and RF were 89.76% and 90.17% respectively (ROC curve in Figure 5).

Mantzaris et al [6] predicted osteoporosis using multi-layer perceptron (MLP) and probabilistic neural network (PNN). Age, sex, height and weight were the input variables to the classifier. They reported a prognosis rate on the testing data of 84.9%. One the same disease, we reported prognosis rate for RF and SVM of 87.85% and 84.68% respectively, and AUC for RF and SVM of 87.08% and 88.05% respectively.

One of the important features of RF is the computation of the variable importance of each variable. We used MeanDecreaseGini (Equations 5 and 6) measure to achieve the variable importance (Table 4). Variable importance tells us which variables out of the 262 variables have strong association with the prediction results. Mantzaris et al.
[6] mentioned that osteoporosis is more prevalent in people older than 50 and occurs in women more than men and that agrees with the first and forth important variables (age and sex) reported by RF (Table 4). lists the top four important variables for the eight diseases.

Table 4 - Top four most importance variable for the eight disease categories

| Disease                | Variable | Variable 2 | Variable 3                  |
|------------------------|----------|------------|-----------------------------|
| Breast cancer          | Age      | Sex        | Secondary malignant         |
| Diabetes no complication | Age      | Hypertension | Hyperlipidemia             |
| Diabetes with/complication | Age      | Diabetes no cm | Hypertension            |
| Hypertension           | Age      | Diabetes no cm | Diabetes no cm             |
| Coronary Atherosclerosis | Age      | Hypertension | Hyperlipidemia             |
| Peripheral             | Age      | Coronary Atherosclerosis | Hypertension   |
| Other Circulatory Diseases | Age      | Dysthymia   | Anemia                      |
| Osteoporosis           | Age      | Race       | Hypertension               |

**Result set II: Sampling vs. non-sampling with random forest**

In this section we show that RF with sampling outperforms standalone RF on the HCUP data set (Table 5). We tested RF with sampling and without sampling. RF with sampling has higher ROC curves and reaches a detection rate of 100% faster than the standalone RF. For demonstration purposes, we included the comparisons for three disease categories, breast cancer, other circulatory diseases and peripheral atherosclerosis (ROC curve in Figure 6, Figure 7, Figure 8).
Table 5 - RF (sampling vs. non-sampling) performance in terms of AUC on eight disease categories

| Disease                        | Sampling  | Non-Sampling |
|--------------------------------|-----------|--------------|
| Breast cancer                  | 0.9017    | 0.8202       |
| Diabetes no complication       | 0.868     | 0.8515       |
| Diabetes with/complication     | 0.9335    | 0.9199       |
| Hypertension                   | 0.8975    | 0.8957       |
| Coronary Atherosclerosis       | 0.923     | 0.9164       |
| Peripheral Atherosclerosis     | 0.9069    | 0.8841       |
| Other Circulatory Diseases     | 0.7846    | 0.7595       |
| Osteoporosis                   | 0.8708    | 0.8232       |

Figure 6 - ROC curve for breast cancer (sampling vs. non-sampling)

ROC curve for breast cancer comparing RF with the sampling and non-sampling approach, average AUC over the 100 runs is 89.76% and 82.02% respectively.
Figure 7 - ROC curve for other circulatory diseases (sampling vs. non-sampling)

ROC curve for other circulatory diseases comparing RF with the sampling and non-sampling approach, average AUC over the 100 runs is 78.46% and 75.95% respectively.
Figure 8 - ROC curve for peripheral atherosclerosis (sampling vs. non-sampling)

ROC curve for peripheral atherosclerosis comparing RF with the sampling and non-sampling approach, average AUC over the 100 runs is 90.69% and 88.41% respectively.

Conclusions and Future Work
In this paper we presented a method for predicting disease risks for patients based on the HCUP data set. Our method can be used in various domains such as health risk management, tailored health communication and medical decision making. For example, a list of possible future diagnosis can be provided to the physician such as appropriate medical tests can be ordered before the disease onset. Moreover, early discussion of lifestyle adjustments may lead to disease prevention. Finally, early
diagnosis and prevention may lead to a decrease in healthcare spending and an increase in care quality.

Our results show that the use of repeated random sub-sampling is useful when dealing with highly unbalanced data. While incorporating demographic information increased the accuracy of the prediction, the increase was not as big as we expected. The variable importance is useful in determining the interaction among variables and the association between the variables and the prediction results. However, the usefulness and the accuracy of our proposed method cannot be exploited without expert medical knowledge. Only physicians can provide insight on how and when to use the proposed method.

Also, a better data set can open the doors for new ideas. Limitations of the HCUP data set such as the arbitrary order of the ICD-9 codes and lack of patient identification lead to rough disease prediction model. For example, if the codes were listed in chronological order according to the date they were diagnosed, one can use different techniques such as HMM to perform disease prediction and association. In addition, the HCUP data set does not provide anonymous patient identifier, which can be used to check if multiple records belong to the same patient and to eliminate the time interval between two diagnoses. However, we believe that the developed method can be improved when used in patient data warehouses where the above information is available.

Overall, the results obtained with the proposed method were satisfactory. The accuracy achieved in disease prediction is comparable or better than the previously
published results. Also, the average AUC of about 88.58% may be acceptable in many applications.

Authors' contributions
Mohammed Khalilia is the main author of this paper. He designed and wrote the code, ran the experiments, collected the results and analyzed them, and drafted the manuscript. Professor Sounak Chakraborty was of great help, he contributed to the design of the experiments, specially the repeated random sub-sampling approach and reviewed the manuscript. Mihial Popescu, Mohammed Khalilia's advisor, provided the data, helped in designing the experiments, provided continuous feedback on the paper, mainly the result section and reviewed the manuscript.

Acknowledgements
The authors would like to thank the University of Missouri Bioinformatics Consortium for providing us with a Supercomputer account to run the experiments.

References

1. Fuster, V., et al., Medical Underwriting for Life Insurance. 2008, McGraw-Hill's AccessMedicine.
2. Yi, T. and Z. Guo-Ji. The application of machine learning algorithm in underwriting process. in Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on. 2005.
3. Cohen, E., et al., Cancer coverage in general-audience and black newspapers. Health Communication, 2008. 23(5): p. 427-435.
4. Moturu, S.T., W.G. Johnson, and L. Huan. Predicting Future High-Cost Patients: A Real-World Risk Modeling Application. in Bioinformatics and Biomedicine, 2007. BIBM 2007. IEEE International Conference on. 2007.
5. Davis, D., et al., Predicting individual disease risk based on medical history. 2008.
6. Mantzaris, D.H., G.C. Anastassopoulos, and D.K. Lymberopoulos. Medical disease prediction using Artificial Neural Networks. in BioInformatics and BioEngineering, 2008. BIBE 2008. 8th IEEE International Conference on. 2008.
7. Hebert, P., et al., Identifying persons with diabetes using Medicare claims data. American Journal of Medical Quality, 1999. 14(6): p. 270.
8. Yu, W., et al., *Application of support vector machine modeling for prediction of common diseases: the case of diabetes and pre-diabetes*. BMC Medical Informatics and Decision Making, 2010. 10(1): p. 16.

9. Zhang, W., et al., *A Comparative Study of Ensemble Learning Approaches in the Classification of Breast Cancer Metastasis*, in Bioinformatics, Systems Biology and Intelligent Computing, 2009. IJCBS '09. International Joint Conference on. 2009. p. 242-245.

10. Project, H.C.a.U. (2009) *Overview of the Nationwide Inpatient Sample (NIS)*.

11. Provost, F., *Machine learning from imbalanced data sets 101*, in Proceedings of the AAAI’2000 Workshop on Imbalanced Data Sets. 2000.

12. Japkowicz, N. and S. Stephen, *The class imbalance problem: A systematic study*. Intelligent Data Analysis, 2002. 6(5): p. 429-449.

13. Quinlan, J. *Bagging, boosting, and C4. 5*. 1996.

14. Breiman, L., *Random forests*. Machine learning, 2001. 45(1): p. 5-32.

15. Chen, C., A. Liaw, and L. Breiman, *Using random forest to learn imbalanced data*. Berkeley (California): University of California Berkeley, 2004.

16. Breiman, L. and others, *Manual-Setting Up, Using, and Understanding Random Forests V4. 0*. At ftp://ftp.stat.berkeley.edu/pub/users/breiman, 2003.

17. Breiman, L., et al., *Classification and regression trees*. Wadsworth. Inc., Belmont, CA, 1984. 358.

18. Hastie, T., et al., *The elements of statistical learning: data mining, inference and prediction*. 2009. 605-622.

19. Bjoern, M., et al., *A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data*. BMC Bioinformatics. 10.

20. Palmer, D., et al., *Random forest models to predict aqueous solubility*. J. Chem. Inf. Model, 2007. 47(1): p. 150-158.

21. Liaw, A. and M. Wiener. *Classification and Regression by randomForest*. 