HHS Public Access

Author manuscript

Int J Med Inform. Author manuscript; available in PMC 2019 April 01.

Published in final edited form as:

Int J Med Inform. 2018 April ; 112: 59–67. doi:10.1016/j.ijmedinf.2018.01.007.

Federated learning of predictive models from federated Electronic Health Records

Theodora S. Brisimi[1], **Ruidi Chen**[1], **Theofanie Mela**[3], **Alex Olshevsky**[1], **Ioannis Ch. Paschalidis**[1,2,*], and **Wei Shi**[1,4]

[1]Department of Electrical & Computer Engineering, and Division of Systems Engineering, Boston University, 8 Saint Mary's St., Boston, MA 02215

[2]Department of Biomedical Engineering, Boston University, 44 Cummington Mall, Boston, MA 02215

[3]Electrophysiology Lab/Arrhythmia Service, Massachusetts General Hospital, 55 Fruit St., Boston, MA 02114

[4]School of Electrical & Computer Engineering, Arizona State University, Tempe, AZ

Abstract

Background—In an era of "big data," computationally efficient and privacy-aware solutions for large-scale machine learning problems become crucial, especially in the healthcare domain, where large amounts of data are stored in different locations and owned by different entities. Past research has been focused on centralized algorithms, which assume the existence of a central data repository (database) which stores and can process the data from all participants. Such an architecture, however, can be impractical when data are not centrally located, it does not scale well to very large datasets, and introduces single-point of failure risks which could compromise the integrity and privacy of the data. Given scores of data widely spread across hospitals/individuals, a decentralized computationally scalable methodology is very much in need.

Objective—We aim at solving a binary supervised classification problem to predict hospitalizations for cardiac events using a distributed algorithm. We seek to develop a general decentralized optimization framework enabling multiple data holders to collaborate and converge to a common predictive model, without explicitly exchanging raw data.

*Corresponding author: Mailing Address: Department of Electrical and Computer Engineering, Boston University, 8 Saint Mary's Street, Boston, MA 02215, USA, Tel: (617) 353-0434, Fax: (617) 353-6440, yannisp@bu.edu.

AUTHORS' CONTRIBUTIONS

W.S. and T.S.B. co-designed and analyzed the methods. T.S.B. and R.C. co-wrote the manuscript, performed the analysis, and produced results and figures. T.M. advised on heart related diseases and helped interpret the results. A.O. co-led the study, co-designed the methods, and commented on the manuscript. I.C.P. led the study, co-designed the methods, and co-wrote the manuscript.

Conflict of Interest Statement:

The authors have no financial or personal relationships with other people or organizations that could inappropriately influence (bias) their work.

Methods—We focus on the soft-margin l_1-regularized sparse Support Vector Machine (sSVM) classifier. We develop an iterative cluster Primal Dual Splitting (cPDS) algorithm for solving the large-scale sSVM problem in a decentralized fashion. Such a distributed learning scheme is relevant for multi-institutional collaborations or peer-to-peer applications, allowing the data holders to collaborate, while keeping every participant's data private.

Results—We test cPDS on the problem of predicting hospitalizations due to heart diseases within a calendar year based on information in the patients Electronic Health Records prior to that year. cPDS converges faster than centralized methods at the cost of some communication between agents. It also converges faster and with less communication overhead compared to an alternative distributed algorithm. In both cases, it achieves similar prediction accuracy measured by the Area Under the Receiver Operating Characteristic Curve (AUC) of the classifier. We extract important features discovered by the algorithm that are predictive of future hospitalizations, thus providing a way to interpret the classification results and inform prevention efforts.

Keywords

predictive models; hospitalization; heart diseases; distributed learning; Electronic Health Records (EHRs); federated databases

INTRODUCTION

Motivation

As the volume, variety, velocity and veracity (the four V's) of the clinical data grow, there is greater need for efficient computational models to mine these data. Insights from these techniques could help design efficient healthcare policies, detect disease causes, provide medical solutions that are personalized and less costly, and finally, improve the quality of care for the patients. We are motivated by problems in the medical domain that can be formulated as binary supervised classification problems and solved using Support Vector Machines; the applications range from prediction of the onset of diabetes,[1,2] prediction of hospitalizations for cardiac events,[3] prediction of medication adherence in heart failure patients,[4] and cancer diagnosis,[5] to automated recognition of the obstructive sleep apnea syndrome.[6]

Results in the literature suggest that sparse classifiers (i.e., those that rely on few features), have strong predictive power and generalize well out-of-sample,[7,8] providing at the same time interpretability in both models and results. Interpretability is crucial for healthcare practitioners to trust the algorithmic outcomes. Another major concern, especially in the medical domain, is the privacy of the data, attracting recent research efforts.[9–11] Two well-known examples of privacy breaches are the Netflix Prize and the Massachusetts Group Insurance Commission (GIC) medical records database. In both cases, individuals were identified even though the data had been through a de-identification process. This demonstrated that one's identity and other sensitive information could be compromised once a single center has access and processes all the data. Especially under the Precision Medicine Initiative,[12] in the near future, these data could include individuals' genome information, which is too sensitive to be shared.

We are particularly interested in addressing three challenges tied to healthcare data: (1) data reside in different locations (e.g., hospitals, doctors' offices, home-based devices, patients' smartphones); (2) there is a growing availability of data, which makes scalable frameworks important; and (3) aggregating data in a single database is infeasible or undesirable due to scale and/or data privacy concerns. In particular, even though maintaining all data in a central location enables the implementation of anonymization measures (e.g., k-anonymity [13]), it introduces a single point of attack or failure and makes it possible for a data breach to expose identifiable data for many individuals. Furthermore, establishing a central data repository requires significant infrastructure investments and overcoming information governance hurdles such as obtaining permissions for storing and processing data. Instead, a decentralized computational scheme that treats the available data as part of a federated (virtual) database, avoiding centralized data collection, processing, and raw data exchanges, may address the above challenges.

Aim

The focus of this paper is to develop a distributed (federated) method to predict hospitalizations during a target year for patients with heart diseases, based on their medical history as described in their Electronic Health Records (EHRs). The records of each patient may lie with them in the patient's smartphone, or may be stored in the EHR systems of different hospitals. In all cases, the collaboration of different parties (agents) is required to develop a global hospitalization prediction model. We will formulate the problem as a binary supervised classification problem and we will develop a distributed soft-margin ℓ_1-regularized (sparse) Support Vector Machines (sSVM) algorithm. We consider SVMs because they are effective classifiers [14] and perform well in predicting hospitalizations.[3] Furthermore, sparse classifiers can reveal relatively few predictive features and, thus, enable interpretation of the predictions.[15]

Health application

We focus on cardiovascular conditions because they comprise a significant portion of morbidity and mortality, as well as, hospitalization in the U.S. and worldwide. In the fact, in the U.S. alone, more than 30% (equal to $9 billion) of hospitalizations deemed preventable are due to cardiovascular conditions.[16] For many decades, the research interest has been focused on understanding the pathophysiology of these conditions and treating them effectively. The efforts have now shifted to the understanding of the disease process and the early prevention. This goal has obvious public health implications, but also socioeconomic significance. It is well known that preventing the progression of the disease process by intensified follow up and treatment can result in long-term stability and improved survival of the patient. Hospitalization is a well-known negative prognostic factor for cardiovascular disease outcome. One critical step in the effort to halt the disease process is the understanding of the etiology and modifiable risk factors of hospitalization.

Main contributions

We summarize our main contributions below:

- We develop a federated optimization scheme (cPDS) for solving the sparse Support Vector Machine problem. Advantages include scalability and the fact that it avoids raw data exchanges, which is important in healthcare. We also demonstrate that cPDS has improved convergence rate and favorable communication cost compared to various centralized and distributed alternatives.
- We apply our new methodology to a dataset of de-identified Electronic Heart Records from the Boston Medical Center, containing patients with heart-related diseases. Each patient is described by a set of features, including demographics, diagnoses, prior admissions, and other relevant medical history.
- We use cPDS to differentiate between patients that are likely and not likely to be hospitalized within a target year and report and discuss the experimental results.
- The proposed cPDS framework is general and can be applied to any learning problem with a "nonsmooth+nonsmooth" loss function objective. Such problems can be found in machine learning, where we aim to minimize functions with non-smooth regularizers, or in distributed model predictive control.

MATERIAL AND METHODS

Objective and background

We consider a dataset extracted from an EHR system, containing patients' demographic data such as age, gender, and race, physical characteristics such as weight, height, Body Mass Index (BMI), medical history captured by diagnoses, procedures, office visits, and a history of drug prescriptions, all captured by a feature vector $\boldsymbol{\phi}_i \in \mathbb{R}^d$, for each patient $i = 1, \ldots, n$. We are interested in predicting whether or not a patient will be hospitalized in a given year, for instance in the next calendar year from the time the record is being examined. We denote a hospitalization by a label $l_i = +1$, and a non-hospitalization by a label $l_i = -1$. Using machine learning terminology, this is a binary classification problem. Using the popular Support Vector Machine (SVM) classifier,[14] we seek to find a hyperplane that maximizes the margin ("distance") between the two classes, while allowing a few points to be misclassified (as shown in Figure 1). Further requiring that a few features are used, we end up with a sparse Support Vector Machine (sSVM) problem:

$$\min_{\boldsymbol{\beta},\beta_0} \sum_{i=1}^{n} h_i(\boldsymbol{\beta},\beta_0) + 0.5\tau\|\boldsymbol{\beta}\|_2^2 + \rho\|\boldsymbol{\beta}\|_1, \quad (1)$$

where $(\boldsymbol{\beta}, \beta_0)$, $\boldsymbol{\beta} \in \mathbb{R}^d$, $\beta_0 \in \mathbb{R}$, identifies the hyperplane/classifier; $h_i(\boldsymbol{\beta},\beta_0 = max\left\{0, 1 - l_i(\phi_i^\top \boldsymbol{\beta} + \beta_0)\right\}$ is a hinge loss function for sample i; τ and ρ are penalty coefficients; $\top$ denotes transpose, and the l_1-norm term $\|\boldsymbol{\beta}\|_1$ serves to induce sparsity.

In the distributed context, we are interested in a setting where each agent[1] holds a part of the data/samples, namely, a subset of $\{\boldsymbol{\phi}_i, i = 1 \ldots, n\}$ and $\{l_i, i = 1, \ldots, n\}$, and would like to collaborate with others to solve Problem (1) for $\boldsymbol{\beta}$ and β_0. Due to scalability, regulatory, and

privacy reasons, agents are not willing to share their raw data with each other or with a processing center. We will develop a decentralized algorithm that avoids raw data exchanges.

Related literature

Problem (1) involves minimization of the sum of two convex but non-smooth terms, i.e., the loss function and the penalty terms $0.5\tau\|\boldsymbol{\beta}\|_2^2+\rho\|\boldsymbol{\beta}\|_1$. When all the data are stored and computations are executed in a centralized unit, we can solve the problem using the interior point (also referred to as barrier) method or the classical subgradient method (SubGD).[17] Another approach with $0(1/\sqrt{k})$ convergence rate that can solve the sSVM and allows a decentralized implementation is the incremental subgradient method (IncrSub).[18] However, IncrSub needs to deploy vanishing step size to reach exact convergence and only works over networks with a ring structure. A recent fully decentralized scheme that has made a significant improvement over the IncrSub is the linear time-average consensus optimization algorithm (LAC).[19] The LAC algorithm is an iterative algorithm that takes smalls steps towards the optimal solution at each iteration utilizing a fixed but small step size and is shown to have $0(1/\sqrt{k})$ convergence rate. A good feature of the LAC is that it improves algorithmic scalability in the size of the network.

The Cluster Primal Dual Splitting

Next, we introduce the general decentralized primal-dual splitting scheme we have designed for solving "nonsmooth+nonsmooth" optimization problems like (1).

Let us assume there is a network of agents, each of which is holding part of the data and they all collectively would like to solve (1) utilizing all data. We consider two scenarios: each agent is holding multiple samples (semi-centralized); or one sample (fully-decentralized). In our healthcare context, agents in the first scenario are hospitals that process the data of their patients only and exchange messages with other hospitals to jointly solve (1). In the second scenario, each patient maintains personal data (e.g.. in a smartphone) and exchanges messages with other patients to jointly solve (1). A combination of these two scenarios is also possible. In either scenario, the m agents are connected through a communication network, which is modeled by an undirected graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$, where $\mathcal{V}=\{1,2,\ldots,m\}$ is the vertex set and $\mathcal{E}$ is the edge set. Throughout the paper, we make the assumption that (A1) the graph $\mathcal{G}$ is connected; and (A2) information exchange happens only between neighbors.

In the decentralized environment, Problem (1) can be reformulated into the following m-cluster splitting formulation:

$$\begin{aligned} \underset{\{\mathbf{x}_j,\mathbf{y}_j\},\forall_j}{min} \quad & \textstyle\sum_{j=1}^{m}\{\mathbf{g}_j(\mathbf{x}_j)+\mathbf{f}_j(\mathbf{y}_j)\} \\ \text{s.t.} \quad & \boldsymbol{\Gamma}_j(\mathbf{A}_j\mathbf{x}_j-\mathbf{y}_j)=0,\ \forall \text{j}, \\ & \mathbf{x}_1=\mathbf{x}_2=\cdots=\mathbf{x}_m, \end{aligned} \tag{2}$$

[1]We will use the generic term "agent" to represent each data/computation center. The term could refer to institutions (hospitals), or even individuals' (patients') devices such as phones, sensors, etc.

where each agent j holds n_j samples (such that $n=\sum_{j=1}^{m} n_j$) and maintains its own copy of the model parameters to be estimated $\mathbf{x}_j=(\boldsymbol{\beta}_j, \beta_{j_0}) \in \mathbb{R}^{d+1}$. Let us define a vector $\mathbf{x} = [\mathbf{x}_1; \ldots ; \mathbf{x}_m] \in \mathbb{R}^{m(d+1)}$ to compactly represent all the local copies; a vector variable $\mathbf{y} = [\mathbf{y}_1; \ldots ; \mathbf{y}_m] \in \mathbb{R}^n$ where each block $\mathbf{y}_j=[y_{j1}; \ldots ;y_{jn_j}] \in \mathbb{R}^{n_j}$ is handled by agent j, and $y_{ji} = l_{ji}(\boldsymbol{\varphi}_{ji}^T \boldsymbol{\beta}_j + \beta_{j0})$, with l_{ji}, $\boldsymbol{\varphi}_{ji}$ being the label and features of the i-th sample held by agent j, respectively. This relationship between $\mathbf{x}_j$ and $\mathbf{y}_j$ is described by the first set of constraints, with $\boldsymbol{\Gamma}_j$ some locally produced/tuned pre-conditioner to ensure the fast convergence of the cPDS algorithm. The function $(\mathbf{f}_j)$ contains all the hinge loss functions at agent j while the function $\mathbf{g}_j(\mathbf{x}_j)$ includes the regularizers over agent j. The consensus among agents is achieved through the second group of constraints in (2). To solve the cluster sSVM, or its more general form (2), in a decentralized fashion, we propose the following algorithm (cPDS), which finds the separating hyperplane through updates on parameters of both the primal problem (2), i.e., $\mathbf{x}_j$, $\mathbf{y}_j$, and its corresponding dual problem, i.e., $\mathbf{q}_j$, $\boldsymbol{\lambda}_j$. The matrices $\boldsymbol{\Gamma}_j$ and $\boldsymbol{\Theta}_j$ serve as algorithmic parameters that accelerate the convergence, and $\boldsymbol{W} = [w_{ij}]$ is a doubly stochastic weight matrix of the graph (see definition in the Appendix). In the algorithm, the norm $\|\boldsymbol{x}\|_{\Theta_j}$ is the $\boldsymbol{\Theta}_j$-weighted norm of $\boldsymbol{x}$ defined as $\|\boldsymbol{x}\|_{\Theta_j}=\sqrt{\boldsymbol{x}'\boldsymbol{\Theta}_j\boldsymbol{x}}$ and denotes the neighboring nodes of node j. We emphasize that the update of the classifier $\boldsymbol{x}_j$ for each agent is implemented using only the local information, which constitutes a salient advantage of cPDS compared to other methods in the literature.

Algorithm 1

The cluster Primal Dual Splitting Algorithm (cPDS).

Cluster PDS Method

INPUT: $\forall j$,

Prepare data/objectives $\mathbf{f}_j$ and $\mathbf{g}_j$. Set parameters $\boldsymbol{\Gamma}_j$ and $\boldsymbol{\Theta}_j$.

INITIALIZE: $\forall j$,

$$\mathbf{x}_j^0 \in \mathbb{R}^{d+1}, \mathbf{y}_j^0 \in \mathbb{R}^{n_j}, \mathbf{q}_j^{-1}=\mathbf{0}, \mathbf{q}_j^0=\boldsymbol{\Gamma}_j(\mathbf{A}_j^\top \mathbf{x}_j^0 - \mathbf{y}_j^0), \boldsymbol{\lambda}_j^{-1}=\mathbf{0} \text{ and } \mathbf{x}_j^0=\mathbf{y}_j^0 - \sum_{i\in\mathcal{N}_j\cup\{j\}} w_{ji}\mathbf{x}_i^0.$$

REPEAT

x – update (locally): $\forall j$

$$\mathbf{x}_j^{k+1}=\arg\min_{\mathbf{x}_j} \left\{ \langle 2\mathbf{q}_j^k - \mathbf{q}_j^{k-1}, \boldsymbol{\Gamma}_j\mathbf{A}_j\mathbf{x}_j\rangle + \mathbf{g}_j(\mathbf{x}_j) + \langle 2\boldsymbol{\lambda}_j^k - \boldsymbol{\lambda}_j^{k-1}, \mathbf{x}_j\rangle + 0.5\|\mathbf{x}_j - \mathbf{x}_j^k\|^2_{\boldsymbol{\Theta}_j} \right\}$$

y- update (locally): $\forall j$

$$\mathbf{y}_j^{k+1}=\arg\min_{\mathbf{y}_j} \left\{ \mathbf{f}_j(\mathbf{y}_j) + \langle \mathbf{q}_j^k, -\boldsymbol{\Gamma}_j\mathbf{y}_j\rangle + 0.5\|\mathbf{y}_j - \mathbf{A}_j\mathbf{x}_j^{k+1}\|^2_{\boldsymbol{\Gamma}_j^\top\boldsymbol{\Gamma}_j} \right\}$$

q-update (locally): $\forall j$

$$\mathbf{q}_j^{k+1}=\mathbf{q}_j^k+\boldsymbol{\Gamma}_j\left(\mathbf{A}_j\mathbf{x}_j^{k+1} - \mathbf{y}_j^{k+1}\right)$$

$\boldsymbol{\lambda}$-update (requires information exchange): $\forall j$

Cluster PDS Method

$$\boldsymbol{\lambda}_j^{k+1} = \boldsymbol{\lambda}_j^k + \mathbf{x}_j^{k+1} - \sum_{i \in \mathcal{N}_j \cup \{j\}} w_{ji} \mathbf{x}_i^{k+1}.$$

UNTIL specific criteria are met.

Theoretical comparison of methods

The table below shows comparative results that illustrate the trade-offs between different methods when applied to the sSVM problem. "Per iteration complexity" measures how many scalar multiplications are needed per iteration. "ε-accuracy iterations" measures how many iterations are needed to reach ε-accuracy. The details for applying cPDS to sSVM can be found in the Appendix.

Performance evaluation

We split the patients in our dataset into two sets: a training and a test set. Training patients are used to train the algorithm. In order to evaluate the accuracy of the predictions, we use the trained classifier to predict the label of patients (whether or not will be hospitalized) in the test set. We measure the performance of cPDS in terms of the Area Under the Receiver Operator Characteristic (ROC) curve (AUC), which plots the true positive rate (i.e., out of the hospitalized patients how many were correctly predicted as hospitalized) versus the false positive rate (i.e., out of the non-hospitalized patients how many were wrongly predicted to be hospitalized). The true positive rate is also referred to as sensitivity or recall and specificity is used to refer to one minus the false positive rate. We also report the computation time, i.e., the cumulative time needed at all nodes to train the model, and the communication cost, which is defined as 2 times the product of the number of edges, the dimension of the features, and the number of iterations. (Only the coefficients of the features are exchanged with adjacent nodes.) Note that the number of edges in the graph decides the amount of information exchanged.

DATA-RESULTS-DISCUSSION

Data description and preprocessing

The data used for the experiments come from the Boston Medical Center and consist of Electronic Health Records (EHRs) containing the medical history in the period 2001–2012 of patients with at least one heart-related diagnosis between 2005–2010. The medical history of each patient includes demographics, diagnoses, procedures, vitals, lab tests, tobacco use, emergency room visits, and past admission records. For each patient, we set a specific target year and we predict hospitalization during that year based on the prior medical history. We follow the steps below to preprocess the data:

- *Setting the target time interval to a calendar year.* Based on preliminary experiments, we observed that there is greater variability in the results when trying to predict hospitalizations in periods of time shorter than a year. Thus, we have designed our experiment to predict hospitalizations in the target time

interval of a year (January 1st – December 31st). We elected to use a calendar year after observing that hospitalizations occur roughly uniformly within a year.

- *Selection of the target year*. As a result of the nature of the data, the two classes (hospitalized and non-hospitalized patients) are highly imbalanced. To increase the number of hospitalized patient examples, if a patient had only one hospitalization throughout 2005–2010, the year of hospitalization will be set as the target year. If a patient had multiple hospitalizations, a target year between the first and the last hospitalizations will be randomly selected. 2010 is set as the target year for patients with no hospitalization, so that there is as much available history for them as possible. By this policy, the ratio of hospitalized patients in the data set is 16.97%.
- *Summarization of the features in the history of a patient*. An effective way to summarize each patient's medical history is to form four time blocks for each medical factor with all corresponding records summarized over one, two, three years before the target year and a fourth time block containing averages of all earlier records. This produces a 215-dimensional vector of features characterizing each patient.
- *Removing patients with no record*. Patients who have no records before the target year are removed, since there is nothing on which a prediction can be based. The total number of patients left is 45,579.
- *Splitting the data into a training set and a test set randomly*. As is common in supervised learning, the population of patients is randomly split into a training and a test set. Since from a statistical point of view, all the data points (patients' features) are drawn from the same distribution, we do not differentiate between patients whose records appear earlier in time than others with later time stamps. A retrospective/prospective approach appears more often in the medical literature and is more relevant in a clinical trial setting, rather than in our algorithmic approach. What is critical in our setting is that for each patient prediction we make (hospitalization/non-hospitalization in a target year), we only use that patient's information before the target year.
- *Normalization of the features*. All predictors are standardized before fed into our algorithm.
- *Balancing the training set.* During training, we oversample the positive class in order to make the two classes balanced.

Experimental results and discussion

The data are distributed between m hospitals connected through a specific graph topology. The cPDS algorithm is considered to converge if the normalized residual, which is defined as the l_2 norm of the difference between the cPDS and the sSVM parameter estimates, is small enough. Since we do not know the true parameter values, the solution from solving sSVM with a centralized (barrier) method is used as a substitute for the ground truth. We want to investigate the impact of two factors on the convergence of cPDS:

1. The number of hospitals $m \in \{5,10\}$.
2. The graph topology: (a) random graph generated by the Erdős–Rényi model, where two nodes are connected with a probability p; (b) cyclic graph, where nodes are connected in a closed chain; (c) fully connected graph, where each node is connected with every other node in the graph.

Table 2 shows the comparison between cPDS and the centralized barrier method, the SubGD, the IncrSub descent and the LAC scheme. For SubGD and IncrSub, we use the steplength rule for the diminishing stepsize[2]. We defined in Materials and Methods the various performance metrics we use. AUC for all methods is similar since they solve the same problem. Just to provide a baseline, we note that using a classifier based on a common risk factor used by cardiologists yields less accurate predictions. Specifically, using the 10-year risk factor for cardiovascular disease developed by the Framingham heart study [20], and comparing that risk-factor to a threshold in order to classify, yields an AUC of 0.56.

The computation cost reported in Table 2 reflects effort at all nodes (or the single node for centralized schemes), so it depends on both the number of iterations and the number of nodes (hospitals). On one hand, the more hospitals there are, the longer the computation time. On the other hand, the more edges in the graph, the less time needed for convergence because information reaches all nodes faster. The communication cost measures the number of messages exchanged between nodes (each message is a vector in $\mathbb{R}^{d+1}$) and is mainly impacted by the number of edges in the graph, which also depends on the number of nodes. In the table, LAC is much more costly than cPDS because it uses a fully distributed approach with n nodes, whereas cPDS uses a graph with $m \ll n$ nodes.

Table 3 considers only cPDS and shows the convergence time and AUC for different combinations of m and graph topology. Fully connected graphs have the most edges, and thus the highest communication cost. But the number of iterations needed for convergence is not significantly smaller than others. In general, the more edges there are, the faster the algorithm converges, since the information exchange becomes faster[3]. We note that when the number of edges is "large enough," the number of iterations needed for convergence stays stable, in other words, the convergence speed comes to be saturated. This is incarnated in the random graph topology. When $m = 5$, 4 edges lead to saturation; and for $m = 10$, 13 edges are needed.

In Table 4 we summarize the important features identified by the cPDS algorithm. We run cPDS for each of the six settings of Table 3, averaged the coefficients corresponding to the various features (elements of the vector $\boldsymbol{\beta}$) over the six runs, and report the features with the largest average coefficients. Note that all features are standardized, and thus it is reasonable to identify important features based on the magnitudes of the average coefficients.

[2]Following the steplength rule, the diminishing stepsize in k-th iteration is set as $a_k = a_0 / \left(\|\tilde{\nabla} \boldsymbol{g}(\boldsymbol{x}^k)\| + \varepsilon \right)$, where a_0 is an initial value of the stepsize and ε a very small number.
[3]Here, we define the convergence speed via the number of iterations needed for convergence.

It is interesting that the classifier identifies many of the diagnoses and health events that are major public health problems and which constitute common reasons for hospitalization with major economic implications. Hypertension, increasing in parallel with obesity, coronary artery disease, as it is identified indirectly by cardiac catheterization, heart failure – a true epidemic as the population is getting older, as well as, cardiac arrest are some of the most prevalent heart-related diagnoses. It is, therefore, important to establish the ability of these diagnoses to predict hospitalization and use such predictions as a tool to prevent the disease process.

CONCLUSIONS

In this paper, we focused on developing a federated learning model that is able to predict future hospitalizations for patients with heart-related diseases using EHR data spread among various data sources/agents. Our proposed decentralized framework, the cluster Primal Dual Splitting (cPDS) algorithm, can solve the sparse Support Vector Machine problem, which yields classifiers using relatively few features and facilitates the interpretability of the classification decisions. cPDS has improved convergence rate compared to various alternatives we present. The method is applicable to any binary classification problem with distributed data.

A major advantage of our formulation is the flexibility to address a range of settings, from fully-centralized to fully-decentralized. We formulate our motivating healthcare problem as a binary classification problem. Information processing can happen either at the level of the patients, e.g., through their smartphones, or at the level of the hospitals that process data of their own patients. cPDS is a general framework and can be applied to any problem that has the structure of minimizing two non-smooth terms. A possible extension of this work could be the analysis of cPDS when the graph that connects the agents is time-varying.

Acknowledgments

We would like to thank Bill Adams and the Boston Medical Center for providing access to the data and valuable advice. We also thank Henghui Zhu and Taiyao Wang for helping accessing and processing the data.

The research has been partially supported by the NSF under grants IIS-1237022, IIS-1724990, CNS-1645681, and CCF-1527292, by the ARO under grant W911NF-12-1-0390, by the NIH under grant 1UL1TR001430 to the Clinical & Translational Science Institute at Boston University, and by the Boston University Digital Health Initiative.

APPENDIX

The sparse Support Vector Machine (sSVM) Problem

sSVM finds the classifier $(\boldsymbol{\beta}, \beta_0)$, $\boldsymbol{\beta} \in \mathbb{R}^d$, $\beta_0 \in \mathbb{R}$, by solving the following problem:

$$\begin{aligned} &\min_{\beta,\beta_0} 0.5\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{n}\xi_i + \kappa\|\boldsymbol{\beta}\|_1 \\ s.t \quad &\xi_i \geq 0, \forall i,\ l_i\left(\phi_i^\top \boldsymbol{\beta} + \beta_0\right) \geq 1 - \xi_i, \forall i. \end{aligned} \tag{3}$$

The $\|\cdot\|_1$ constraint in the above formulation is forcing the classifier $\boldsymbol{\beta}$ to be sparse. In the decentralized setting with m agents, problem (3) could be reformulated into the following m-cluster splitting form:

$$\begin{aligned} &\min_{\boldsymbol{\beta},\beta_0} \sum_{j=1}^{m} \left\{ \sum_{i=1}^{n_j} [1 - y_{ji}]_+ + 0.5\tau_j \|\boldsymbol{\beta}_j\|_2^2 + \rho_j \|\boldsymbol{\beta}_j\|_1 \right\} \\ \text{s.t.} \quad &\gamma_{ji}(l_{ji}(\phi_{ji}^{\top}\boldsymbol{\beta}_j + \beta_{j0}) - y_{ji}) = 0, \forall j, i; \boldsymbol{\beta}_1 = \boldsymbol{\beta}_2 = \cdots = \boldsymbol{\beta}_m; \beta_{1,0} = \beta_{2,0} = \cdots = \beta_{m0}, \end{aligned}$$

where each agent (hospital) j holds n_j samples (such that $n = \sum_{j=1}^{m} n_j$), and $y_{ji} = l_{ji}(\phi_{ji}^{\top}\boldsymbol{\beta}_j + \beta_{j0})$. The parameters γ_{ji}'s are arbitrary nonzero scalar constants and serve as algorithmic parameters.

Definition 1

[Doubly stochastic matrix] $\mathbf{W} = [w_{ij}]$ is defined to be a doubly stochastic matrix generated by the following the Metropolis rule on $\mathcal{G}$:

$$w_{ij} = \begin{cases} \frac{1}{max\{\text{degree}(i), \text{degree}(j)\} + 1}, & \text{if} (i,j) \in \varepsilon, \\ 0, & \text{if} (i,j) \notin \varepsilon \text{ and } i \neq j, \\ 1 - \sum_{k \in V} w_{ik}, & \text{if } i = j. \end{cases}$$

Such rule allows each agent i to generate w_{ij}, $\forall j$, by only using local information (its own and neighbors' degree information). Note we always have $-\mathbf{I}_m \prec \mathbf{W} \preccurlyeq \mathbf{I}_m$. Let us also define $\mathrm{L} \triangleq (\mathbf{I}_m - \mathbf{W}) \otimes \mathbf{I}_{d+1}$ and $U \triangleq \sqrt{L}$. Here, $\otimes$ denotes the Kronecker product of matrices and $\mathbf{I}_m$ the $m \times m$ identity matrix. We note $\mathbf{U}$ has the same null space as that of $\mathbf{L}$.

Insights on the cPDS Algorithm

To get some insight on how the algorithm works, let us further write (2) into an even more compact form.

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{y}} \ \{\mathbf{g}(\mathbf{x}) + \mathbf{f}(\mathbf{y})\} \\ &\text{s.t.} \ \ \boldsymbol{\Gamma}(\mathbf{A}\mathbf{x} - \mathbf{y}) = 0, \ \mathbf{U}\mathbf{x} = 0. \end{aligned} \quad (4)$$

We note that $\mathbf{U}\mathbf{x} = 0$ is equivalent to $\mathbf{x}_1 = \mathbf{x}_2 = \cdots = \mathbf{x}_m$ as long as the graph is connected.

We will make two more assumptions: (A3) The functions $\mathbf{g}$: $\mathbb{R}^{m(d+1)} \to \mathbb{R}$ and $\mathbf{f}$: $\mathbb{R}^n \to \mathbb{R}$ are both proper, closed, and convex. (A4) The solution set χ^* is nonempty and bounded. Assumption (A3) imposes a minimal requirement on the objectives to conduct convex analysis. Assumption (A4) is obviously satisfied by the sSVM problem.

The augmented Lagrangian function of (4) is as follows

$$\mathcal{L} = \mathbf{g}(\mathbf{x}) + \mathbf{f}(\mathbf{y}) + \langle \mathbf{r}, \mathbf{U}\mathbf{x} \rangle + \langle \mathbf{q}, \boldsymbol{\Gamma}(\mathbf{A}\mathbf{x} - \mathbf{y}) \rangle,$$

where **r** and **q** contain the dual variables. The idea behind our proposed algorithm is based on minimizing the Lagrangian function with respect to the primal variables **x** and **y** and the dual variables **r** and **q.** However, when doing so, the **x** – update is not implementable in a fully distributed setting. This is the key limitation that cPDS is addressing and contributing to the literature.

Application of cPDS on ℓ_1-Regularized Support Vector Machines

We will show the details of applying the cPDS framework to solve the distributed sSVM problem. Assume that n samples of data are distributed among m agents that want to collectively agree on a global classifier to separate the two classes. Each agent is holding n_j samples and maintains a copy ($\boldsymbol{\beta}_j$, β_{j0}) of the classifier parameters to be estimated. ($\boldsymbol{\beta}_j$, β_{j0}) are updated in each iteration of the method, using data locally stored at the agent j as well as information that the agent receives from its neighbors. Let $\boldsymbol{\phi}_{ji} \in \mathbb{R}^d$ and $l_{ji} \in \mathbb{R}$ be the features and the label of sample i in agent j accordingly, and f_{ji} be the corresponding hinge loss for that sample. $\boldsymbol{g}_j$ contains the regularizers of parameters ($\boldsymbol{\beta}_j$, β_{j0}) for each agent j. Define $\mathbf{a}_{ji} = (l_{ji}\boldsymbol{\phi}_{ji}, l_{ji})$, which we will use later. In every iteration each agent updates $\mathbf{x}_j = (\boldsymbol{\beta}_j, \beta_{j0}) \in \mathbb{R}^{d+1}$, $\boldsymbol{y}_j \in \mathbb{R}^{n_j}$, $\boldsymbol{q}_j \in \mathbb{R}^{n_j}$ and $\boldsymbol{\lambda}_j \in \mathbb{R}^{d+1}$. Let us illustrate below the cPDS updates that each agent is performing. For simplicity, in the implementation, we use $\Theta_j = \theta_j \mathbf{I}_{d+1}$ where θ_j is a positive scalar maintained by agent j locally. Next we describe the updates over each agent j.

x – update

$$
\begin{aligned}
&(\boldsymbol{\beta}_j^{k+1}, \beta_{j0}^{k+1}) \\
&\quad = \underset{\boldsymbol{\beta}_j, \beta_{0j}}{\operatorname{arg\,min}} \sum_{i=1}^{n_j} \gamma_{ji} (2q_{ji}^k \\
&\qquad - q_{ji}^{k-1}) \boldsymbol{a}_{ji}^\top \binom{\boldsymbol{\beta}_j}{\beta_{j0}} \\
&\qquad + \frac{\tau}{2} \|\boldsymbol{\beta}_j\|^2 + \rho \|\boldsymbol{\beta}_j\|_1 + (2\boldsymbol{\lambda}_j^k - \boldsymbol{\lambda}_j^{k-1})^\top \binom{\boldsymbol{\beta}_j}{\beta_{j0}} \\
&\qquad + 0.5 \left\| \begin{pmatrix} \boldsymbol{\beta}_j \\ \beta_{j0} \end{pmatrix} - \begin{pmatrix} \boldsymbol{\beta}_j^k \\ \beta_{j0}^k \end{pmatrix} \right\|^2 \theta_j .
\end{aligned}
\tag{5}
$$

The simple form of the non-smooth $\mathbf{g}_j$ allows us to get a closed form solution for this problem. Problem (5) can be decoupled into two problems, one that finds $\boldsymbol{\beta}_j^{k+1}$, whose solution is given by the soft thresholding function, i.e., $\forall t = 1, \ldots, d$,

$$
\beta_{jt}^* = \operatorname{sgn}(u_{jt})(|u_{jt}| - \mu)_+ = \begin{cases} u_{jt} - \mu, & \text{if } u_{jt} > \mu, \\ 0, & \text{if } |u_{jt}| \le \mu, \\ u_{jt} + \mu, & \text{if } u_{jt} < -\mu, \end{cases}
$$

with = $\frac{2\rho}{\tau+\theta_j}$, and $\mathbf{u}_j = -\frac{1}{\tau+\theta_j}[\sum_{i=1}^{n_j}\boldsymbol{\gamma}_{ji}l_{ji}(2q_{ji}^k - q_{ji}^{k-1})\phi_{ji} + (2\boldsymbol{\lambda}_{j,1:d}^k - \boldsymbol{\lambda}_{j,1:d}^{k-1}) - \theta_j\boldsymbol{\beta}_j^k]$ and one that finds β_{j0}^{k+1}, which has as an optimal solution:

$$\beta_{j0}^* = \frac{-\sum_{i=1}^{n_j}\gamma_{ji}l_{ji}(2q_{ji}^k - q_{ji}^{k-1}) - (2\boldsymbol{\lambda}_{i,d+1}^k - \boldsymbol{\lambda}_{i,d+1}^{k+1}) + \theta_j\beta_{j0}^k}{\theta_j}.$$

y- update

$$\mathbf{y}_j^{k+1} = \underset{\mathbf{y}_j}{\arg min}\sum_{i=1}^{n_j}\left\{max\{0, 1-y_{ji}\} + \langle\gamma_{ji}q_{ji}^k, -y_{ji}\rangle + \frac{1}{2}||\gamma_{ji}(l_{ji}\boldsymbol{\phi}_{ji}^\top\boldsymbol{\beta}_j^{k+1} + l_{ji}\beta_{j0}^{k+1} - y_{ji})||^2\right\}.$$

To deal with the second non-smooth term, the hinge loss function, we consider three cases for each term: $1 - y_{ji} > 0$, $1 - y_{ji} < 0$, $1 - y_{ji} = 0$. For each agent *j*, we can obtain every entry of $\mathbf{y}_j$ in parallel, i.e., for all *i*:

- Solve

$$\tilde{y}_{ji}^{k+1} = \underset{\mathbf{y}_{ji}}{\arg min}\left\{\frac{\gamma_{ji}^2}{2}y_{ji}^2 + (-1 - \gamma_{ji}q_{ji}^k - \gamma_{ji}^2l_{ji}\phi_{ji}^\top\boldsymbol{\beta}_j^{k+1} - \gamma_{ji}^2l_{ji}\beta_{j0}^{k+1})\right\} \Rightarrow \tilde{y}_{ji}^{k+1} = \frac{1}{\gamma_{ji}^2}(1 + \gamma_{ji}q_{ji}^k + \gamma_{ji}^2l_{ji}\phi_{ji}^\top\boldsymbol{\beta}_j^{k+1} + \gamma_{ji}^2l_{ji}\beta_{j0}^{k+1}).$$

  If $1 - \tilde{y}_{ji} > 0$, then $y_{ji}^{k+1} = \tilde{y}_{ji}^{k+1}$; otherwise proceed to the next step.

- Solve

$$\tilde{y}_{ji}^{k+1} = \underset{\mathbf{y}_{ji}}{\arg min}\left\{\gamma_{ji}q_{ji}^k, -y_{ji} + \frac{1}{2}\|\gamma_{ji}(l_{ji}\phi_{ji}^\top\boldsymbol{\beta}_j^{k+1} + l_{ji}\beta_{j0}^{k+1} - y_{ji})\|^2\right\} \Rightarrow \tilde{y}_{ji} = \frac{1}{\gamma_{ji}^2}(\gamma_{ji}q_{ji}^k + \gamma_{ji}^2l_{ji}\phi_{ji}^\top\boldsymbol{\beta}_j^{k+1} + \gamma_{ji}^2l_{ji}\beta_{j0}^{k+1}).$$

  If $1 - \tilde{y}_{ji}^{k+1} < 0$, then $y_{ji}^{k+1} = \tilde{y}_{ji}^{k+1}$; otherwise proceed to the next step.

- $y_{ji}^{k+1} = 1$.

q-update: ∀*i*

$$q_{ji}^{k+1} = q_{ji}^k + \gamma_{ji}(l_{ji}\boldsymbol{\phi}_{ji}^\top\boldsymbol{\beta}_j^{k+1} + l_{ji}\beta_{j0}^{k+1} - y_{ji}^{k+1}).$$

λ-update

$$\lambda_j^{k+1} = \lambda_j^k + \sum_{i \in \mathcal{N}_j \cup \{j\}} w_{ji} \mathbf{x}_j^{k+1}.$$

Last, let us mention that the storage needed to operate cPDS for sSVM following the above updates is O(nd), which is the same as the other methods listed in Table 1.

References

1. Kumari VA, Chitra R. Classification of diabetes disease using support vector machine. Int J Eng Res Appl. 2013; 3:1797–1801.
2. Yu W, Liu T, Valdez R, Gwinn M, Khoury MJ. Application of support vector machine modeling for prediction of common diseases: the case of diabetes and pre-diabetes. BMC Med Inform Decis Mak. 2010; 10:16. [PubMed: 20307319]
3. Dai W, Brisimi TS, Adams WG, Mela T, Saligrama V, Paschalidis IC. Prediction of hospitalization due to heart diseases by supervised learning methods. Int J Med Inf. 2015; 84:189–197.
4. Son YJ, Kim HG, Kim EH, Choi S, Lee SK. Application of support vector machine for prediction of medication adherence in heart failure patients. Healthc Inform Res. 2010; 16:253–259. [PubMed: 21818444]
5. Statnikov A, Tsamardinos I, Dosbayev Y, Aliferis CF. GEMS: a system for automated cancer diagnosis and biomarker discovery from microarray gene expression data. Int J Med Inf. 2005; 74:491–503.
6. Khandoker AH, Palaniswami M, Karmakar CK. Support vector machines for automated recognition of obstructive sleep apnea syndrome from ECG recordings. IEEE Trans Inf Technol Biomed. 2009; 13:37–48. [PubMed: 19129022]
7. Hastie T, Tibshirani R, Wainwright M. Statistical learning with sparsity: the lasso and generalizations, CRC press. 2015
8. Chen R, Paschalidis IC. Outlier Detection Using Distributionally Robust Optimization under the Wasserstein Metric. 2017 ArXiv Prepr. ArXiv170602412.
9. Dwork C, Roth A, et al. The algorithmic foundations of differential privacy. Found Trends® Theor Comput Sci. 2014; 9:211–407.
10. Huang, Z., Mitra, S., Vaidya, N. Differentially private distributed optimization. Proc. 2015 Int. Conf. Distrib. Comput. Netw., ACM, 2015; p. 4http://dl.acm.org/citation.cfm?id=2684480
11. Nozari, E., Tallapragada, P., Cortes, J. Differentially private distributed convex optimization via functional perturbation. IEEE Trans Control Netw Syst. 2016. http://ieeexplore.ieee.org/abstract/document/7577745/
12. Collins FS, Varmus H. A new initiative on precision medicine. N Engl J Med. 2015; 372:793–795. [PubMed: 25635347]
13. Sweeney L. k-anonymity: A model for protecting privacy. Int J Uncertain Fuzziness Knowl-Based Syst. 2002; 10:557–570.
14. Cortes C, Vapnik V. Support-vector networks. Mach Learn. 1995; 20:273–297.
15. Xu, T., Brisimi, TS., Wang, T., Dai, W., Paschalidis, IC. A joint sparse clustering and classification approach with applications to hospitalization prediction. Decis. Control CDC 2016 IEEE 55th Conf. On, IEEE; 2016. p. 4566-4571.http://ieeexplore.ieee.org/abstract/document/7798964/
16. Jiang, HJ., Russo, CA., Barrett, ML. Nationwide Frequency and Costs of Potentially Preventable Hospitalizations, 2006. US Agency for Healthcare Research and Quality; Rockville, MD: 2010. HCUP Statistical Brief# 72http://www.hcupus.ahrq.gov/reports/statbriefs/sb72.jsp
17. Bertsekas, DP. Nonlinear programming, Athena scientific Belmont. 1999. http://www.academia.edu/download/6133373/30954470x.pdf

18. Nedic A, Bertsekas DP. Incremental subgradient methods for nondifferentiable optimization. SIAM J Optim. 2001; 12:109–138.

19. Olshevsky, A. Linear time average consensus on fixed graphs and implications for decentralized optimization and multi-agent control. 2014. ArXiv Prepr ArXiv14114186https://arxiv.org/abs/1411.4186

20. D'Agostino RB, Vasan RS, Pencina MJ, Wolf PA, Cobain M, Massaro JM, Kannel WB. General cardiovascular risk profile for use in primary care. Circulation. 2008; 117:743–753. [PubMed: 18212285]

Research Highlights

- A new federated learning framework is proposed that can learn predictive models through peer-to-peer collaboration of federated databases without raw data exchanges.
- Using the EHR, it is possible to accurately predict heart-related hospitalizations.
- The predictive model derived improves prediction accuracy over existing risk metrics.
- The predictive model is sparse, identifying the most informative EHR variables for hospitalization prediction.

SUMMARY TABLE

What was already known	What this study added to our knowledge
Electronic Health Records (EHRs) can potentially be used to assess a person's health and predict future hospitalizations. Some comprehensive risk metrics exist (Framingham risk factor) to assess the risk of a major heart-related episode.	Using the entirety of the EHR, it is possible to accurately predict an individual's hospitalization for cardiac events in the following calendar year, improving upon the accuracy of existing risk metrics (such as the Framingham risk factor).
Centralized machine learning methods are typically used to train predictive models (classifiers) from data.	A new distributed learning framework has been developed to solve the learning (classification) problem in a setting where data reside with many agents, no raw data get exchanged, and the agents collaborate to jointly learn the model (classifier). The distributed algorithms is more scalable than centralized algorithms or earlier distributed methods.
	The new learning framework is flexible to accommodate a range of data aggregation levels at the nodes, from each node holding a single data point (e.g., an individual EHR) to a setting where each node maintains many data points (a hospital maintaining all the hospital's EHRs).
	The sparse classifiers produced by our method automatically concentrate on relatively few features, facilitating the interpretability of the classification results.

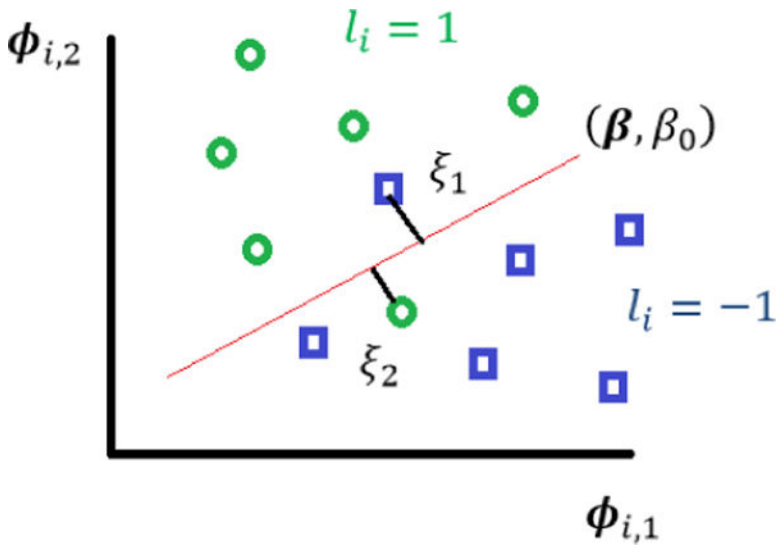

Figure 1.
Support Vector Machines.

Table 1

Theoretical comparison of various methods that solve the sSVM problem.

Method	Decentralized?	Per iteration complexity	ϵ-accuracy iterations
Subgradient Descent	×	O(nd)	$O(1/\epsilon^2)$
Incremental Subgradient	×	O(d)	$O(1/\epsilon^2)$
Linear Average Consensus (LAC)	√	$O(n^2 + nd)$	$O(1/\epsilon^2)$
Cluster Primal Dual Splitting (cPDS)	√	$O((n + m^2)d)$	$o(1/\epsilon)$

Table 2

Experimental comparison for various methods that solve the sSVM problem.

Method	Distributed?	AUC	Number of iterations	Computation cost (sec)	Communication cost
Subgradient Descent	×	0.7667	1500	2055	N/A
Barrier	×	0.7688	32	40,174	N/A
Incremental Subgradient	×	0.7734	554	6.3485	N/A
Linear Average Consensus (LAC)	√	0.7683	200	27,703	1.04e+11
Cluster Primal Dual Splitting (cPDS) (m = 10, random graph)	√	0.7806	100	544	2600

Table 3

Performance of cPDS for different number of agents m and different graph topologies.

Graph topology	Number of hospitals	Number of edges	Number of iterations	Computation time (sec)	Communication cost	AUC
Random	5	4	90	906	720	0.7738
Random	10	13	100	544	2600	0.7806
Cyclic	5	5	90	909	900	0.7736
Cyclic	10	10	250	1482	5000	0.7723
Fully connected	5	10	90	917	1800	0.7696
Fully connected	10	45	100	585	9000	0.7747

Table 4

Important features.

	Important features	**Average coefficients over 6 runs**
	Age	224.03
Factors - 1 year the before target year	Diagnosis of Heart Failure- 1 year before the target year	225.63
	Admission due to Other Circulatory System Diagnoses- 1 year before the target year	204.14
	Admission due to Heart Failure- 1 year before the target year	183.09
	Admission due to a Percutaneous Cardiovascular Procedure- 1 year before the target year	168.53
	Admission for Cardiac Defibrillator Implant with Cardiac Catheterization – 1 year before the target year	145.78
	Admission due to Cardiac Arrest- 1 year before the target year	144.96
	Systolic Blood Pressure Measured -1 year before the target year	136.71
Factors - 2 years before the target year	Diagnosis of Heart Failure- 2 years before the target year	162.20
	Admission due to Other Circulatory System Diagnoses- 2 years before the target year	139.12
	Admission due to Cardiac Arrest- 2 years before the target year	127.59
	Admission due to a Circulatory Disorder Except Acute Myocardial Infarction -2 years before the target year	184.57
	Admission due to Cardiac Valve or Other Major Cardiothoracic Procedure- 2 years before the target year	175.84
	Diagnostic ultrasound of heart - 2 years before the target year	137.40
	Admission for Acute and Subacute endocarditis - 2 years before the target year	129.07
	Admission for Acute Myocardial Infarction - 2 years before the target year	120.73
Factors - 3 years before the target year	Diagnosis of Heart Failure- 3 years before the target year	178.02
	Admission due to Other Circulatory System Diagnoses- 3 years before the target year	158.67
	Admission due to a Percutaneous Cardiovascular Procedure- 3 years before the target year	170.34
	Admission due to Cardiac Arrest-3 years before the target year	155.05
	Admission for Acute and Subacute endocarditis - 3 years before the target year	135.91