RESEARCH ARTICLE

Multilevel Weighted Support Vector Machine for Classification on Healthcare Data with Missing Values

Talayeh Razzaghi[1], Oleg Roderick[2], Ilya Safro[1]*, Nicholas Marko[2]

1 School of Computing, Clemson University, Clemson, SC, United States of America, **2** Department of Data Science, Geisinger Health System, Danville, PA, United States of America

* isafro@clemson.edu

Abstract

This work is motivated by the needs of predictive analytics on healthcare data as represented by Electronic Medical Records. Such data is invariably problematic: noisy, with missing entries, with imbalance in classes of interests, leading to serious bias in predictive modeling. Since standard data mining methods often produce poor performance measures, we argue for development of specialized techniques of data-preprocessing and classification. In this paper, we propose a new method to simultaneously classify large datasets and reduce the effects of missing values. It is based on a multilevel framework of the cost-sensitive SVM and the expected maximization imputation method for missing values, which relies on iterated regression analyses. We compare classification results of multilevel SVM-based algorithms on public benchmark datasets with imbalanced classes and missing values as well as real data in health applications, and show that our multilevel SVM-based method produces fast, and more accurate and robust classification results.

Introduction

Modern healthcare can be characterized as *personalized*, *evidence-driven* and *model-assisted* [1]. As healthcare industry is becoming more integrated with data science, planners and practitioners have to continuously choose the best available machine learning methods to use on medical data that is inherently sparse, noisy, and scanned for rare events more often that for the norm.

In the clinical environment, decisions made based on predicting risks and positive outcomes should be ideally supported by statistical learning models. These models can be seen as either a simplified risk-assessment model [2], or a sophisticated machine learning method [3, 4]. In either case, it is based on a query of relevant clinical and operational history. While prediction of real-valued health metric over time is of interest to the healthcare domain, it more properly belongs to the field of medical simulation models.

Due to the nature of big data in healthcare, the role of data analysis, particularly classification methods, is critical to support better decision for personalized medicine, that is, decision-

making with awareness that patients can be classified into groups based on their personal characteristics and the patterns observed in patient-provider-insurer interactions, and that patients from different groups will have different responses to treatment and different risk outcomes. Thus, if we describe the most common task of predictive analytics for healthcare in one informal sentence, it would be: solving classification problems on clinical data using specialized preocessing and specialized predictive algorithms.

Comprehensive medical information comes in multiple categories (that can be stored in multiple databases, with different formats and rules of access). Categories include: biometric information, medical codes referring to clinical transactions, insurance claims and payments, results of laboratory tests, narratives such as doctors' notes, socioeconomic data characterizing life conditions and choices of individual patients, and molecular data (genomics, proteomics, metabolomics). Our work is focused on the categories items placed earlier in this list. They are considered 'traditional medical information', are recorded for millions, rather than thousands of patients, and are best source material for study of large patient populations. Due to considerations of patient privacy, and the proprietary nature of electronic medical records [5], the databases cannot be queried continuously. Every instance of data acquisition and integration is a separate effort that is cost-effective only when the resulting predictive model shows high quality. Thus, progress in evidence-driven healthcare depends on how well state-of-the art algorithms of machine learning are adapted to clinical data.

We note that classical mathematical, and computer science issues, such as scalability, or convergence rate are rarely a major issue for healthcare applications. Instead, an algorithm is ranked based on its ability to process raw medical data, with such problematic features as sparsity, missing entries, noise and imbalanced outputs. Because of the encounter nature of patient-provider interaction, medical data is inherently sparse: when a clinical encounter occurs, the number of and contents of labels attached to it vary widely [6]; outside of an encounter, the state of the patient is unknown. The outcomes of interest in classification problems are imbalanced, because, as a rule, healthcare analytics is motivated by rare events such as healthcare emergencies, severe chronic conditions, gaps and bottlenecks in access to care.

This work was prompted by several projects completed with the Division of Applied Research and Clinical Informatics, Dept. of Data Science; Geisinger Health System. For the first motivating example (Example 1, see Section), we use our 2014 feasibility study [7] of merging insurance information (6 aggregate features, based on the history of claims and payments) together with clinical encounter information (10–20 features chosen by hand from patient biometrics, medications and diagnostic codes). The goal of the initial study was to predict the financial risk for a particular patient (a common metric in insurance practice, derived as a ratio of individual expenses and average expenses for a large demographic group). We attempted to use a standard clustering technique, k-nearest neighbors with empirically selected weighting, to achieve the basic results before we developed the proposed method. Unfortunately, the results were very unsatisfactory: the chance of mis-categorization (in one-against all binary classification) was close to 50% for all risk groups. Intuitive explanations such as "some patients have entered a high-risk state that is not yet reflected in their financial information" could not be formally verified or used to explain the poor performance, which prompted interest in using the more advanced machine learning methods.

For Example 2 (see Section), we use our preliminary investigation of patients' response to public outreach [8], such as annual flu awareness campaigns. We included basic demographic and clinical information on patients targeted by 35 identically organized campaigns. (the features included in the data were: age, sex and BMI of the patient, and binary variables identifying whether the patient was assigned the most commonly occurring medication codes and prescribed the most commonly occuring medications). The data was used to build a model

predicting whether a given patient is likely to respond to the reminder, or to choose not to get vaccinated, or use a different provider. Again, our initial core predictive model was standard: logistic regression with empirically selected weighting of training data which is widely used in healthcare informatics.

We intend to show that in each case, the predictive models are made more effective with the use of an advanced machine learning algorithm developed with awareness of sparsity and class skewness (imbalance) in data.

Methods

Our study was approved by Geisinger's Institutional Review Board. Information from individual electronic medical records was de-identified prior to use in the study.

Support vector machines (SVM) are among the most well-known optimization-based supervised learning methods, originally developed for binary classification problems [9]. The main idea of SVM is to identify a decision boundary with maximum possible margin between the data points of each class. Training nonlinear SVMs is often a time consuming task when the data is large. This problem becomes extremely sensitive when the model selection techniques are applied. Requirements of computational resources, and storage are growing rapidly with the number of data points, and the dimensionality, making many practical classification problems less tractable. In practice, when solving SVM, there are several parameters that have to be tuned. Advanced methods, such as the grid search and the uniform design for tuning the parameters, are usually implemented using iterative techniques, and the total complexity of the SVM strongly depends on these methods, and on the quality of the employed optimization solvers such as [10].

In this paper, we focus on SVMs that are formulated as the convex quadratic programming (QP). Usually, the complexity required to solve such SVMs is between $\mathcal{O}(n^2)$ to $\mathcal{O}(n^3)$ [11]. For example, the popular QP solver implemented in LibSVM [10] scales between $\mathcal{O}(n_f n_s^2)$ to $\mathcal{O}(n_f n_s^3)$ subject to how efficiently the LibSVM cache is employed in practice, where n_f, and n_s are the numbers of features, and samples, respectively.

Typically, the gradient descent methods achieve good performance results on such models, but still tend to be very slow for large-scale data (when effective parallelism is hard to achieve). Several works have recently addressed this problem. Parallelization usually splits the large set into smaller subsets and then performs a training to assign data points into different subsets [12]. In [11], a parallel version of the Sequential Minimal Optimization (SMO) was developed to accelerate the solution of QP. Although parallelizations over the full data sets often gain good performance, they can be problematic to implement due to the dependencies between variables, which increases communication. Moreover, although specific types of SVMs might be appropriate for parallelization (such as the Proximal SVM [13]), the question of their practical applicability for high-dimensional datasets still requires further investigation. Another approach to accelerate the QP is chunking [14, 15], in which the optimization problem is solved iteratively on the subsets of training data until the global optimum is achieved. The SMO is among the most popular methods of this type [16], which scales down the chunk size to two vectors. Shrinking to identify the non-support vectors early, during the optimization, is another common method that significantly reduces the computational time [10, 14, 17]. Such techniques can save substantial amounts of storage when combined with caching of the kernel data. Digesting is another successful strategy that "optimizes subsets of training data closer to completion before adding new data" [18]. *Summarizing computational and EMR problems mentioned above, we note that being highly flexible and parametrizable to be applied on a variety of complex manifolds, applications of SVMs on large-scale healthcare data without significant decrease in time complexity can be extremely expensive.*

Let us formally describe a supervised classification problem on data consisting of real-valued variables and categorical variables converted into binaries. Given a training set $\mathcal{J} = \{(x_i, y_i)\}_{i=1}^{l}$, that is a set of data points with known labels, where $(x_i, y_i) \in \mathbb{R}^{n+1}$, and l and n are the numbers of data points and features, respectively, and $y_i \in \{-1, 1\}$ denotes the class label for each data point i in $\mathcal{J}$. We denote by $\mathbf{C}^-$ and $\mathbf{C}^+$, the "majority" (points with $y_i = -1$) and "minority" (points with $y_i = +1$) classes respectively such that $\mathcal{J} = \mathbf{C}^+ \cup \mathbf{C}^-$.

Support Vector Machines

The optimal SVM classifier is determined by the parameters w and b through solving the convex optimization problem

$$\min \quad \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{l} \xi_i \tag{1}$$

$$\text{s.t.} \quad y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i \qquad i = 1, \dots, l \tag{2}$$

$$\xi_i \geq 0 \qquad\qquad i = 1, \dots, l \tag{3}$$

where ϕ maps training instances x_i into a higher dimensional space, $\phi : \mathbb{R}^n \to \mathbb{R}^m$ ($m \geq n$). The term slack variables ξ_i ($i \in \{1, \dots, l\}$) in the objective function is used to penalize misclassified points. This approach is also known as *soft margin* SVM. The magnitude of penalization is controlled by the parameter C. Many existing algorithms (such as SMO, and its implementation in LIBSVM tool [10] that we use) solve the Lagrangian dual problem instead of the primal formulation, which is a popular strategy due to its faster and more reliable convergence.

Weighted Support Vector Machines

Imbalanced classification tasks (when the sizes of classes are very different) are another major problem that, in practice, can lead to poor performance measures [19]. Imbalanced learning is a significant emerging problem in many areas, including medical diagnosis [20–22], face recognition [23], bioinformatics [24], risk management [25, 26], and manufacturing [27]. Many standard SVM algorithms often tend to misclassify the data points of the minority class. One of the most well-known techniques to deal with imbalanced data is the cost-sensitive learning (CSL). The CSL addresses imbalanced classification problems through different cost matrices. The adaptation of cost-sensitive learning with the regular SVM is known as *weighted support vector machine* (WSVM [28], also termed as Fuzzy SVM) [29]. The main idea is to consider weighting scheme in learning such that the WSVM algorithm builds the decision hyperplane based on the relative contribution of data points in training. In contrast to the standard SVM, the penalization costs are different for the positive ($\mathbf{C}^+$) and negative ($\mathbf{C}^-$) classes:

$$\min \quad \frac{1}{2} \| w \|^2 + C^+ \sum_{\{i | y_i = +1\}}^{n_+} \xi_i + C^- \sum_{\{j | y_j = -1\}}^{n_-} \xi_j \tag{4}$$

$$\text{s.t.} \quad y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i \qquad i = 1, \dots, l \tag{5}$$

$$\xi_i \geq 0 \qquad\qquad i = 1, \dots, l, \tag{6}$$

where C^+, and C^- are the parameters associated with the positive, and negative classes, which

assign different importance weights to each data class. The formulations (1) and (4) are solved through the Karush-Kuhn-Tucker conditions.

The Gaussian kernel function (radial basis function, RBF) defined as

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\gamma \parallel \boldsymbol{x}_i - \boldsymbol{x}_j \parallel^2\right), \gamma \geq 0, \tag{7}$$

is used in the dual formulation of (W)SVMs. This kernel has been confirmed as the most successful for the UCI benchmark in multiple studies. Parameter tuning is required to set optimal or near optimal C, C^+, C^-, and kernel function parameters (e.g. bandwidth parameter for RBF kernel function) to achieve good results for (W)SVM. This process becomes problematic and time-consuming particularly when the size of data is very large. Hence we aim to develop an efficient and effective classification method, called the Multilevel (W)SVM, that is scalable and works with imbalanced healthcare data.

Multilevel Support Vector Machines

The proposed algorithm belongs to the family of multilevel optimization strategies [30] whose goal is to approximate the system at multiple scales of coarseness and to obtain a final solution by combining the information from different scales. The multilevel framework for SVM [31] scales efficiently for large classification problems whose hierarchy of coarser representations is constructed based on the approximated k-nearest neighbors graphs (AkNN). We note that the exact nearest neighbor graph methods are rather computationally expensive due to construction of the k-NN graph structure. There is a lack of exact and scalable nearest neighbor search algorithms with good performance, when data is high-dimensional. Several attempts have been made to propose an approximate search [32, 33], in which not all the neighbors obtained are exact, but still generally close to the exact neighbors. The multilevel support vector machine method (Fig 1) consists of three main phases, namely, coarsening, coarse support vector initial learning, and uncoarsening.

The coarsening phase. The coarsening algorithms are the same for both C^+, and C^-, so we provide only one of them. Given a class of data points C, the coarsening begins with a construction of an approximated k-nearest neighbors (AkNN) graph $G = (V,E)$, where $V = C$, and E are the edges of AkNN. A gradual coarsening of the training set is constructed using fast point selection method [34] in AkNN graph. (In fact, this version of coarsening is a simplified coarsening developed for combinatorial optimization problems on graphs such as in [35, 36].)

The goal is to select a set of representative points $\hat{V}$ for the next-coarser problem, where $|\hat{V}| \geq Q|V|$, and Q is the parameter for the size of the coarse level graph. In practice, it can often be done by selecting a maximal independent set of points such as in [34]. However, we found that ensuring a slightly denser uniform coverage of the points can lead to much better results than finding an independent set of points (nodes in AkNN) as was suggested in [34]. (An independent set is a set of vertices in a graph when no two vertices are connected by an edge.) Thus, we extended the set of coarse points by setting a parameter for the minimum number of points that in our experiments was set to 50% of the fine data points. The second requirement for $\hat{V}$ is that it has to be a dominating set of V. (The dominating set of nodes is a subset of V such that each vertex in V is either in this set or adjacent to one or more vertices in it.)

The coarsening for class C is presented in Algorithm 1. The algorithm consists of several iterations of independent set of V selections that are complementary to already chosen sets. We begin with choosing a random independent set (line 2) using the greedy algorithm. It is eliminated from the graph, and the next independent set is chosen and added to $\hat{V}$ (lines 5–11). For imbalanced cases, when WSVM is used, we avoid of creating very small coarse problems for C^-. Instead, already very small class is continuously replicated across the rest of the hierarchy if
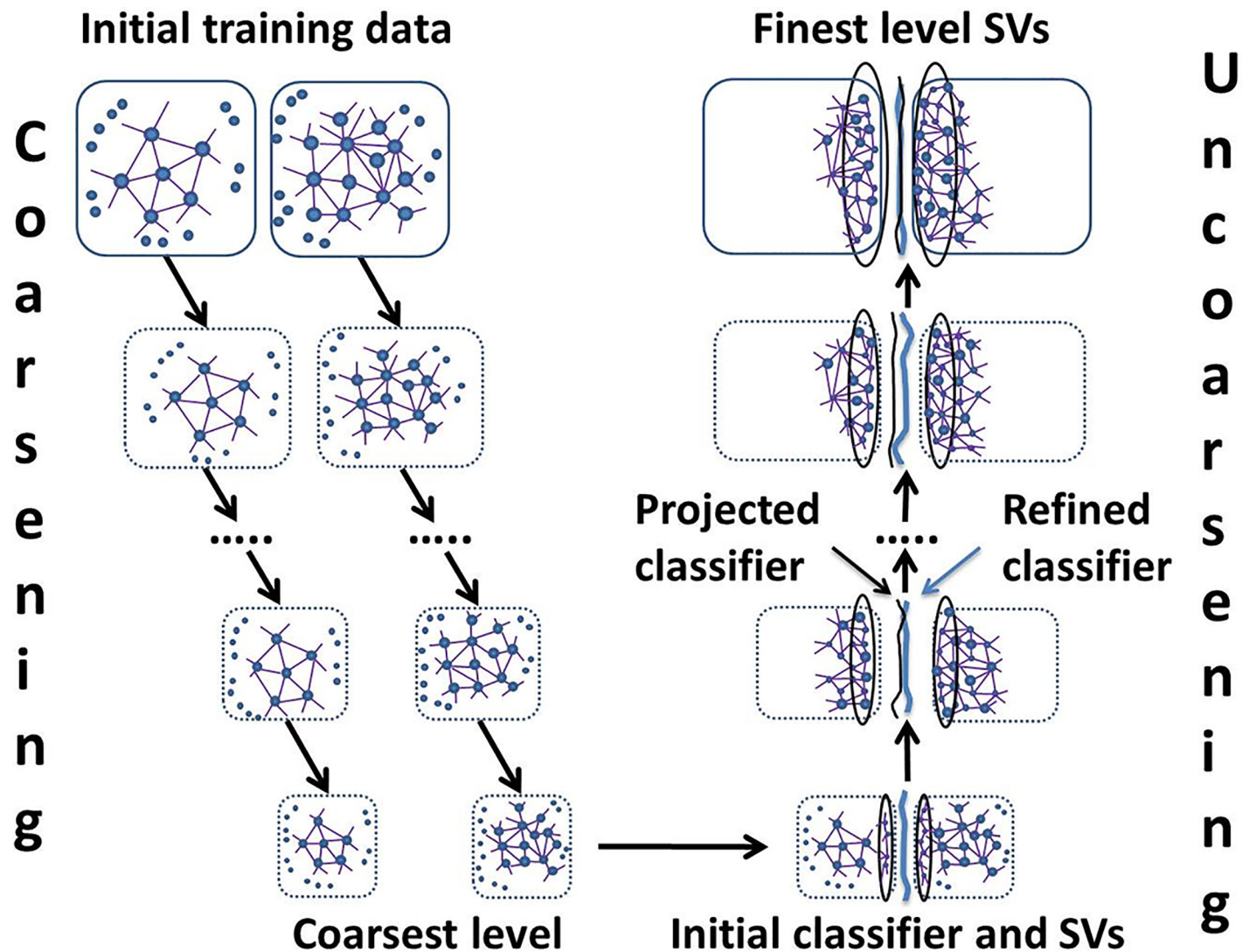
Fig 1. The multilevel SVM framework consists of three phases: gradual training set coarsening, coarsest support vectors' learning, and gradual support vectors' refinement (uncoarsening). Pairs of AkNN graphs correspond to two classes of learning.

$\mathbf{C}^+$ still requires coarsening. We note that this method of coarsening will reduce the degree of skewness in the data and make the data approximately balanced at the coarsest level. The multilevel framework recursively calls the coarsening process until it creates a hierarchy of r coarse representations $\{\mathcal{J}_i\}_{i=0}^r$ of $\mathcal{J}$. At each level of this hierarchy, the corresponding AkNNs' $\{G_i = (V_i, E_i)\}_{i=0}^r$ are saved for future use at the uncoarsening phase. The corresponding data and labels at level i is denoted by $(X_i, Y_i) \in \mathbb{R}^{k \times (n+1)}$, where $|X_i| = k$.

Algorithm 1 The Coarsening
1: **Input:** $G = (V, E)$ for class $\mathbf{C}$
2: $\hat{V} \leftarrow$ select maximal independent set in G
3: $\hat{U} \leftarrow V \setminus \hat{V}$
4: **while** $|\hat{V}| < \mathcal{Q} \cdot |V|$ **do**
5:   **while** $\hat{U} \neq \emptyset$ **do**
6:     randomly pick $i \in \hat{U}$

```
 7:     Û ← Û \ {i}
 8:     Û ← Û \ {neighbors of i in Û}
 9:     V̂ ← V̂ ∪ {i}
10:  end while
11:  Û ← V \ V̂
12: end while
13: return V̂
```

Supervised support vector initial learning.   After the hierarchy is created, the support vectors learning is performed at the coarsest level, where the number of data points is sufficiently small.

At the coarsest level r, when $|\mathcal{J}_r| << \mathcal{J}$, we can apply an exact algorithm for training the coarsest classifier. Typically, the size of the coarsest level depends on the computational resources. However, for the (W)SVM problems, one can also consider some criteria of the separability between $\mathbf{C}_r^+$, and $\mathbf{C}_r^-$ [37], i.e., if a fast separability test exists or additional data properties are available. We used the simplest criterion bounding $\mathcal{J}_r$ to 500. Processing the coarsest level includes an application of the uniform design (UD) [38] model selection to improve the quality of classifiers. The nested UD search is an efficient method used for automatic model selection for SVMs. This method is applied to select the candidate set of parameter combinations and carry out a k-fold cross-validation to evaluate the quality of each parameter combination.

The uncoarsening phase.   Support vectors, and classifier are projected throughout the hierarchy from the coarsest to the finest levels. At each level, a solution to the current fine level is updated and optimized based on the solution of the previous coarse level. The locally optimal support vectors are obtained by gradual refinement of the support vectors projected from the coarse level.

Given the solution of coarse level $i + 1$ (the set of support vectors S_{i+1}, and parameters C_{i+1}, and γ_{i+1}), the primary goal of the refinement is to update and optimize this solution for the current fine level i. Unlike many other multilevel algorithms, in which the inherited coarse solution contains projected variables only, in our case, we initially inherit not only the coarse support vectors (the solution that can represent the whole training set [39, 40]) but also parameters for model selection. This is because the model selection is an extremely time-consuming component of (W)SVM, and can be prohibitive at fine levels. However, at the coarse levels, when the problem is much smaller than the original, we can apply much heavier methods for model selection with almost no loss in the running time of the framework. In particular, at each level of ML(W)SVM, after updating the training set and before running SVM, the UD is performed on the training data. Because the data might be imbalanced, we select the optimal parameter set with respect to the maximum G-mean value. The optimal C_{i+1}, γ_{i+1} of previous level are used as the initial C_i and γ_i at level i, and they will be updated at each level based on the new training set. The C_i and γ_i that result in higher G-mean will be selected as optimal or near-optimal parameters at all levels.

The refinement is presented in Algorithm 2. The coarsest level is solved exactly and reinforced by the model selection (lines 2–5). If i is one of the intermediate levels, we build the set of training data $data_{train}^{(i)}$ by inheriting the coarse support vectors S_{i+1} and adding to them some of their approximated nearest neighbors at level i (lines 6–7) (in our experiments, usually not more than 5). If the size of $data_{train}^{(i)}$ is still small enough (relatively to the existing computational resources, and the initial size of the data) for applying model selection, and solving SVM on the whole $data_{train}^{(i)}$, then we use coarse parameters C_{i+1}, and γ_{i+1} as initializers for the current level, and retrain (lines 9–10,19). Otherwise, the coarse C_{i+1}, and γ_{i+1} are inherited in C_i, and γ_i (line 12). Then, being large for direct application of the SVM, $data_{train}^{(i)}$ is clustered into K clusters,

and pairs of P nearest opposite clusters are retrained, and contribute their solutions to S_i (lines 15–17). The number of K is determined in as

$$K = |data_{train}^{(i)}|/Q_{dt}. \tag{8}$$

We note that cluster-based retraining can be done in parallel, as different pairs of clusters are independent. Moreover, the total complexity of the algorithm does not suffer from reinforcing the cluster-based retraining with model selection.

Algorithm 2 The Refinement at level i
1: **Input:** $\mathcal{J}_i, S_{i+1}, C_{i+1}, \gamma_{i+1}$
2: **if** i is the coarsest level **then**
3:   Calculate the best (C_i, γ_i) using UD
4:   $S_i \leftarrow$ Apply SVM on X_i
5: **end if**
6: Calculate nearest neighbors N_i for support vectors S_{i+1} from the existing AkNN G_i
7: $data_{train}^{(i)} \leftarrow S_{(i+1)} \cup N_i$
8: **if** $|data_{train}^{(i)}| < Q_{dt}$ **then**
9:   $C^O \leftarrow C_{i+1}$; $\gamma^O \leftarrow \gamma_{i+1}$
10:   Run UD using the initial center (C^O, γ^O)
11: **else**
12:   $C_i \leftarrow C_{i+1}$; $\gamma_i \leftarrow \gamma_{i+1}$
13: **end if**
14: **if** $|data_{train}^{(i)}| \geq Q_{dt}$ **then**
15:   Cluster $data_{train}^{(i)}$ into K clusters
16:   $\forall k \in K$ find P nearest opposite-class clusters
17:   $S_i \leftarrow$ Apply SVM on pairs of nearest clusters only
18: **else**
19:   $S_i \leftarrow$ Apply SVM directly on $data_{train}^{(i)}$
20: **end if**
21: **Return** S_i, C_i, γ_i

For imbalanced data, the WSVM can easily be adopted as the base classifier for multilevel framework (MLWSVM). The regular SVM does not perform well on imbalanced data because it tends to train models with respect to the majority class and technically ignores the minority class. However, the effect of imbalanced issue decreases while using multilevel framework since we prevent creating very small coarse sets for the minority class even if the majority class can still be coarsened.

Often, methods for imbalanced classification demonstrate poor performance on data with missing values (such as [41]) that is a frequent situation in healthcare data. Therefore, we apply imputation methods prior the classification model. Such imputation methods have been well studied in statistical analysis and machine learning domains [42–46]. Problems with missing data can be categorized into three types: data is completely at random (MCAR), missing at random (MAR), and not missing at random (NMAR). MCAR occurs while any feature of a data instance is missing completely random and is independent of the values of other features. Data is MAR, when the data instance with missing feature is dependent on the value of one or more of the instances' other features. NMAR occurs when the data instance with missing feature is dependent on the value of the other missing features. Even though MCAR is more desirable, in many real-world problems, MAR occurs frequently in practice [42].

In the imputation methods, the goal is to substitute a missing value with a meaningful estimation [45]. This can be done either directly from the information on the dataset or by

constructing a predictive model for this purpose. Standard methods for imputation are mean imputation [47], kNN imputation [48], Bayesian principal component analysis (BPCA) imputation [49], and the expectation maximization (EM) [50]. We apply the EM method which is one of the most successful imputation methods [51]. The EM method iteratively applies linear regression analysis and fits a new linear to the estimated data until a local optimum is achieved [50, 52]. In the regularized adaption of EM method, the conditional maximum likelihood estimation of regression parameters is replaced in the conventional EM algorithm [53].

Regularized Expectation-Maximization

In our preprocessing, when the data contains many missing values, we apply the EM algorithm. It iteratively calculates the maximum-likelihood (ML) estimates of parameters by exploring the relationship between the complete and incomplete data (with missing features) [54]. In many cases, it has been demonstrated that the EM algorithm achieves a reliable global convergence to a local maximizer (from almost any starting point), and economical storage. It is not computationally expensive, and can be easily implemented [55]. The EM algorithm maximizes the log-likelihood (L) of the incomplete data

$$L(\Theta; \chi) = \sum_{i=1}^{n} \log p(x_i|\Theta), \tag{9}$$

where $\chi = \{x_i | i = 1, \ldots, n\}$ are the observations with independent distribution $p(x)$ parameterized by Θ and P is the distribution function of the complete data given Θ. The regularized EM algorithm (REM) is developed to control the level of uncertainty associated to missing values [56]. The main idea is to regularize the likelihood function according to the mutual relationship between the observations and the missing data with little uncertainty and maximum information. Intuitively, it is desirable to select the missing data that has a high probabilistic association with the observations, which shows that there is little uncertainty on the missing data given the observations. It performs linear regression iteratively for the imputation of missing values. The REM algorithm optimizes the penalized likelihood as follows:

$$\tilde{L}(\Theta; \chi) = L(\Theta; \chi) + \Gamma P(\chi, \Upsilon|\Theta). \tag{10}$$

The trade-off between the degree of regularization of the solution and the likelihood function is controlled by the so-called regularization parameter that is represented by Γ [56]. In addition to reducing the uncertainty of missing data, the REM preserves the advantage of the standard EM method. This method is very efficient for over-complicated models.

The EM algorithm implementation in this paper is based on iterated linear regression analysis. In the regularized EM algorithm, a regularized estimation method substitutes the conditional maximum likelihood estimation of regression parameters in the conventional EM algorithm. We used the modules from [50], which apply truncated total least squares (with fixed truncation parameter) and ridge regression with generalized cross-validation as regularized estimation methods. We only perform the REM imputation for all classification datasets in the paper.

Performance Measures

Classification algorithms are evaluated using the performance measures calculated from the confusion matrix (see Table 1).

Table 1. Confusion matrix.

	Positive class	Negative Class
Positive class	True Positive (TP)	False Positive (FP)
Negative Class	False Negative (FN)	True Negative (TN)

doi:10.1371/journal.pone.0155119.t001

For binary classification problems, the performance measures are defined as sensitivity (SN), specificity (SP), and G-mean, accuracy (ACC), namely,

$$\text{SN} = \frac{TP}{TP + FN}, \quad \text{SP} = \frac{TN}{TN + FP}, \tag{11}$$

$$\text{G-mean} = \sqrt{\text{SP} * \text{SN}}, \tag{12}$$

and

$$\text{ACC} = \frac{TP + TN}{FP + TN + TP + FN}. \tag{13}$$

Numerical Results

Due to proprietary nature of medical data, anonymized medical records can be made available for research purposes, but cannot be shared in open access. For validation and reproduction, it is helpful to first examine the performance of proposed methods on standard public data sets. Our source code is available at [57].

We evaluate the proposed classification framework on public (UCI [58], and the cod-rna dataset [59]), and healthcare proprietary binary classification benchmarks [7, 8]. Both the coarsest and refinement (W)SVM models are solved using LIBSVM-3.18 [10], and the FLANN library [33] is used to create the AkNN graphs. We used a 'composite' algorithm of FLANN, which is a combination of multiple randomized KD trees and hierarchical k-means trees. According to [33], it outperforms separate KD trees and hierarchical k-means trees, so we report only the results of the 'composite' algorithm in FLANN. The number of requested nearest neighbors in AkNN is selected as $k = 10$. Increasing k does not improve the results. We chose P for the number of nearest opposite-class clusters as 10% of the number of clusters in the corresponding class (Algorithm 2, line 16). In particular, after clustering each class individually, instead of training each cluster in one class with all clusters of the other class, we will pick 10% of clusters that belong to the other class, in a way that these clusters are the closest clusters to the current cluster k. Next, we train cluster k with these 10% of nearest opposite clusters. For example, if there are 20 clusters in the majority class and 5 clusters in the minority class, we pick a cluster in the minority class and train it with 2 clusters of the majority class. Multilevel frameworks, data processing and further scripting are implemented in MATLAB 2012a [60]. The C4.5 [61], Naive Bayes (NB) [62], Logistic Regression (LR) [63], and 5-Nearest Neighbor (5NN) [64] are implemented using WEKA [65] interfaced with MATLAB. A typical 10-fold cross validation setup is used. We create missing values on the public data training sets by discarding the features randomly. The misclassification penalty or weights are selected as inversely proportional to the size of each class in our implementation. As a preprocessing step, the whole data is normalized such that it has a zero mean and unitary standard deviation. before classification. The nested uniform design (UD) is performed on the training data as the model

selection for (W)SVM [38]. <mark>The UD methodology is very successful for model selection in supervised learning [66].</mark> The close-to-optimal parameter set is achieved in an iterative nested process [38]. The optimal parameter set is selected based on *G-mean* maximization, since data might be imbalanced. A 9- and 5-point run design is performed for the first and second stages of the nested UD due to its superiority for the UCI data [38], and the performance measures such as sensitivity, specificity, G-mean and accuracy are calculated on the testing data.

The nested UD method is performed in two stages. In the first stage, a 9-run UD sampling pattern is conducted in the appropriate search range for C and γ. In the second stage, the search range for each parameter is fixed around the best point from the first stage. Then a 5-runs UD sampling pattern is searched in the new range. The total number of parameter combinations is 13 (the center point at the second stage is the duplicate point which is trained and should be considered only once). The optimal parameters are determined inside the MLSVM model each time before SVM training each time. The initial range of parameter C is between 0.01 and 100, the initial range of parameter γ is between 0.005000 and 3.000078 for the nested UD. For the REM implementation, we used multiple ridge regression within 5-fold cross-validation. We performed the REM imputation in each fold of cross validation on the training data (90% of the whole set). This means that there will be no transfer of information from the validation data set into the training data through the imputation scheme to avoid biased results.

Public data sets

We compared several methods with the proposed ML(W)SVM to classify data with missing values. We show the comparative results of MLSVM, MLWSVM, SVM, WSVM, Naive Bayes, C4.5, LR, and 5NN algorithms evaluated on public data sets in Table 2. These methods are examined for different missing value ratios (r_{mv}) selected as 5%, 10%, 20%, and 40% (Table 3). We used the REM method for missing data imputation [50]. The best results for their corresponding missing values' levels among all methods are shown in bold which makes clear that MLWSVM and WSVM perform better than the other methods in general for all missing value ratios (see last row in Table 3). In fact, MLWSVM and WSVM result in higher G-mean values in 22 out of 40 dataset-r_{mv} combinations followed by MLSVM, SVM, and C4.5 with 13 out of 40. Moreover, the ML(W)SVM techniques achieve faster computational time in comparison to the standard (W)SVM (Table 4). We note that, while the results on publicly available data are easy to relate to and reproduce, testing on relatively small and well-known data sets has its limitations. Most standard methods and implementations achieve acceptable metrics of quality because have been tested on and tuned using these data sets. In practice, we do not expect that

Table 2. Public data sets.

Dataset	r_{imb}	n_f	$|\mathcal{J}|$	$|C^+|$	$|C^-|$
Twonorm	0.50	20	7400	3703	3697
Letter26	0.96	16	20000	734	19266
Ringnorm	0.50	20	7400	3664	3736
Cod-rna	0.67	8	59535	19845	39690
Clean (Musk)	0.85	166	6598	1017	5581
Advertisement	0.86	1558	3279	459	2820
Nursery	0.67	8	12960	4320	8640
Hypothyroid	0.94	21	3919	240	3679
Buzz	0.80	77	140707	27775	112932
Forest	0.98	54	581012	9493	571519

doi:10.1371/journal.pone.0155119.t002

Table 3. Comparative G-mean results for ML(W)SVM against the regular SVM, WSVM, NB, C4.5, 5NN, and LR on academic datasets for different fractions of missing values (r_{mv}) using the REM imputation method.

Dataset	r_{mv}	MLSVM	MLWSVM	SVM	WSVM	C4.5	5NN	NB	LR
Twonorm	5%	**0.98**	**0.98**	**0.98**	**0.98**	0.86	0.97	**0.98**	**0.98**
	10%	**0.98**	**0.98**	0.97	0.97	0.87	0.97	0.97	0.97
	20%	**0.98**	**0.98**	**0.98**	**0.98**	0.88	0.97	0.97	**0.98**
	40%	0.97	0.97	0.97	0.97	0.89	0.97	**0.98**	**0.98**
Letter	5%	0.97	**1.00**	0.99	0.99	0.97	0.98	0.86	0.81
	10%	0.98	**1.00**	0.98	0.99	0.98	0.98	0.86	0.80
	20%	**1.00**	**1.00**	0.99	0.99	0.97	0.98	0.87	0.80
	40%	0.95	0.97	0.96	**0.99**	0.97	0.98	0.88	0.83
Ringorm	5%	0.97	0.98	0.97	0.98	0.91	0.61	**0.99**	0.76
	10%	0.98	0.98	**0.99**	**0.99**	0.91	0.62	0.98	0.76
	20%	**0.98**	**0.98**	0.97	**0.98**	0.91	0.62	**0.98**	0.76
	40%	**0.98**	**0.98**	0.97	**0.98**	0.91	0.62	**0.98**	0.76
Cod-rna	5%	0.95	**0.96**	**0.96**	**0.96**	0.95	0.92	0.66	0.93
	10%	0.95	**0.96**	0.95	**0.96**	0.95	0.91	0.66	0.92
	20%	0.95	**0.96**	0.95	0.95	0.94	0.91	0.67	0.92
	40%	**0.95**	**0.95**	**0.95**	**0.95**	0.93	0.90	0.68	0.91
Clean	5%	**1.00**	0.99	0.98	**1.00**	0.83	0.92	0.79	0.89
	10%	0.99	**1.00**	0.99	**1.00**	0.83	0.91	0.79	0.89
	20%	**1.00**	**1.00**	**1.00**	**1.00**	0.83	0.91	0.79	0.89
	40%	**1.00**	**1.00**	**1.00**	**1.00**	0.82	0.92	0.79	0.89
Advertisement	5%	0.87	0.87	0.87	0.87	**0.92**	0.81	0.60	0.82
	10%	**0.87**	**0.87**	0.86	0.86	0.86	0.85	0.62	0.82
	20%	0.83	0.85	0.83	0.85	**0.89**	0.83	0.61	0.83
	40%	0.84	0.86	0.87	0.81	**0.91**	0.85	0.62	0.82
Nursery	5%	0.99	0.99	**1.00**	**1.00**	**1.00**	**1.00**	0.00	**1.00**
	10%	0.99	0.99	**1.00**	**1.00**	**1.00**	**1.00**	0.00	**1.00**
	20%	0.96	0.96	**1.00**	**1.00**	**1.00**	**1.00**	0.00	**1.00**
	40%	0.92	0.92	**1.00**	**1.00**	**1.00**	0.99	0.46	**1.00**
Hypothyroid	5%	0.83	0.87	0.81	0.87	0.96	0.76	**0.97**	0.88
	10%	0.85	0.86	0.78	0.86	0.96	0.76	**0.96**	0.89
	20%	0.84	0.86	0.72	0.86	0.96	0.75	**0.97**	0.90
	40%	0.86	0.88	0.84	0.88	0.96	0.76	**0.97**	0.89
Buzz	5%	**0.94**	**0.94**	**0.94**	**0.94**	**0.94**	0.93	0.89	**0.94**
	10%	**0.94**	**0.94**	**0.94**	**0.94**	**0.94**	0.93	0.89	**0.94**
	20%	0.92	**0.94**	0.93	**0.94**	**0.94**	0.93	0.88	0.93
	40%	0.93	0.93	0.93	0.93	**0.94**	**0.94**	0.86	**0.94**
Forest	5%	0.90	**0.91**	0.90	**0.91**	**0.91**	0.87	0.80	0.00
	10%	0.92	**0.93**	0.91	0.92	0.88	0.85	0.78	0.00
	20%	0.91	**0.92**	0.90	0.91	0.89	0.84	0.77	0.00
	40%	0.88	0.89	0.88	**0.90**	0.85	0.82	0.73	0.00
# of bold values		13	22	13	22	13	4	9	10

doi:10.1371/journal.pone.0155119.t003

a specialized method will show a dramatic, consistent improvement in quality. A fast computational time without any loss in quality is the most significant result in this work, which illustrates the advantage of multi-level approach that inherits flexible SVM parameters found at coarser levels.

Table 4. Computational time in seconds (not including the REM method).

	MLSVM	SVM	MLWSVM	WSVM
Twonorm	5	28	5	28
Letter	30	138	32	139
Ringnorm	4	25	4	26
Cod-rna	266	1831	281	1857
Clean	17	95	15	82
Advertisement	98	227	100	231
Nursery	25	187	31	192
Hypothyroid	2	3	2	3
Buzz	2209	25257	2999	26026
Forest	13328	352500	13360	353210

doi:10.1371/journal.pone.0155119.t004

Healthcare data sets

We present the results of comparison of classification algorithms on the real-life healthcare data sets. We show the results on Example 1 (see Section, and Table 5) in Table 6, a classification task of assigning a patient in a correct group by financial risk, which are ordered in ascending manner from group 1 with the lowest level of risk, to group 5 with the highest level of risk. The data used in the study was provided by Geisinger Health System in a follow-up study to the internal report on prediction on integrated clinical and financial data [7].

The motivation behind the original study was to determine how much integration of the medical and financial data changes the outcomes of clustering and classification operations based on financial data alone. In the original study, a logistic regression (LR) (implemented as *mnrfit* in MATLAB) was used as a default binary classifier; other standard choices such as nearest-neighbor or naive Bayesian classifiers were rejected in the original report as producing lower quality of prediction. Our comparison illustrates the point that a specialized method developed for imbalanced, incomplete data here outperforms an approach that is accepted as default for a healthcare application.

We compare the accuracy of commonly used in healthcare data analysis LR with that obtained by ML(W)SVM. The strategy "one-against-all" is used for multi-class classification. This strategy performs training a classifier per class with the data points of that class as positive class and the rest of the data points are trained as negative class. Results in Tables 6–8 have also been obtained using 10-fold cross validation.

To interpret the results, we note that correct identification of intermediate risk categories is a very difficult problem in medical informatics. To our knowledge, there is no good definition of "average health", either evidence-driven or philosophical, that would help an expert to identify such patient features that do not indicate an acute crisis, or an almost certain safety from crisis. Accordingly, it is not surprising that neither approach does well on the risk categories 2–4; there is also not a lot of motivation to improve the model there. On the other hand, it is

Table 5. Healthcare datasets. The set "Example 1" has 10000 observations in each class. In set "Example 2", the majority and minority classes contain 50400, and 33600 observations, respectively. For details about the data see [8].

Data	n_f	$|\mathcal{J}|$	No. of classes
Example 1	16	50000	5
Example 2	13	84000	2

doi:10.1371/journal.pone.0155119.t005

Table 6. Accuracy of financial risk problem with five risk classes (Example 1) using the REM imputation method.

Class	1	2	3	4	5
LR	0.58	0.54	0.53	0.51	0.59
MLSVM	0.83	0.78	0.77	0.78	0.90
MLWSVM	0.86	0.76	0.76	0.77	0.91

doi:10.1371/journal.pone.0155119.t006

important to identify and predict the very low-risk patients (knowing that status ahead of time allows resource re-allocation leading to savings and improved service for everyone) and the very high-risk patients (so that clinical and financial resources could be prepared for the forthcoming crisis).

Accordingly, it is important that the use of an advanced method of machine learning changes the quality of prediction from almost worthless ('toss a coin') to workable (accuracy of 0.7).

In Table 8, we compare results for the widely used basic approach and ML(W)SVM prediction for Example 2 (see Section), a study of patient's response to hospital flu outreach. In this problem, the goal is to find a binary classifier that will predict whether the patient will get vaccinated after reminder, or not (this includes using a different provider for vaccination). In the preliminary study, we used adaptive linear regression model (LASSO for adaptive selection of features, logistic regression on actual prediction).

Response to outreach is not a crucial life-or-death issue, we are performing this study to see if predictive modeling can assist with resource allocation (which patients to contact, how much medical personnel effort to dedicate to outreach and then vaccination). Arguably, accuracy is more important than specificity here. Even the basic results (using linear regression) were met with approval the CPSL (Care Patient Service Line: a division responsible for coordinating efforts of local, small-scale healthcare providers operating under Geisinger). SVM methods (almost 10 percent improvement) provide additional justification for the use of machine learning on merged data to assist planning in clinical practice.

Table 7. Sensitivity, specificity and G-mean of financial risk problem with five risk classes (Example 1) using ML(W)SVM and REM imputation methods.

	MultilevelSVM			MultilevelWSVM		
	SN	SP	G-mean	SN	SP	G-mean
Class 1	0.86	0.73	0.79	0.89	0.74	0.81
Class 2	0.89	0.34	0.55	0.86	0.36	0.56
Class 3	0.89	0.28	0.50	0.88	0.29	0.50
Class 4	0.88	0.40	0.60	0.87	0.40	0.58
Class 5	0.96	0.69	0.81	0.96	0.70	0.82

doi:10.1371/journal.pone.0155119.t007

Table 8. Comparison of Multilevel WSVM against Multilevel SVM and Adaptive Logistic Regression (LR) using the REM imputation method. Improved results are in bold.

	G-mean	SN	SP	ACC
Adaptive LR	0.7516	0.8903	0.6345	0.7619
MLSVM	**0.8012**	**0.9750**	**0.6583**	**0.8496**
MLWSVM	**0.8016**	**0.9739**	**0.6598**	**0.8495**

doi:10.1371/journal.pone.0155119.t008

Discussion

Large-scale data, missing or imperfect features, skewness distribution of classes are common challenges in pattern recognition of many healthcare problems. We have successfully extended a powerful machine learning technique, support vector machines, to the *scalable multilevel framework* of cost-sensitive learning SVM to deal with imbalanced classification problems. Our multilevel framework substantially improves the computational time without losing the quality of classifiers for large-scale datasets. We have shown that MLWSVM produces superior results than MLSVM and the regular SVM methods in most cases. This work can be extended to tackle other classification problems with large-scale imbalanced data (combined from different sources) with missing features in healthcare and engineering applications.

From the perspective of evidence-driven healthcare, our work shows that application of cutting edge machine learning techniques (in this case, fast multilevel classifiers) makes enough of a difference to justify the additional development effort for typical examples from clinical practice. While the improvements in precision and specificity we show in this study are both under 10% and are modest in general perspective, the result in healthcare is significant.

To our knowledge, such complex combined behavioral/operational phenomena as inference of financial risk from medical history (Example 1), or prediction of effectiveness of public outreach (Example 2), don't have a satisfactory casual explanation. The classical (1990s) clinical practice offered two equally unsatisfactory options: not having a capability for prediction at all, or relying on very basic statistical techniques (based on a single data source, with very high rate of false-positive classification outcomes). The existing mature models (such as actuarial projections of financial risk) do not benefit from integration of data from multiple sources, and may, in fact, turn out to be ineffective outside of their scope in patient population and metrics of interest (as we have shown in [7]). Thus, in the modern clinical practice we have to rely on newly developed machine learning tools, tuned on data from multiple sources. Thus, our work can also be extended to handle other classification problems on massive, multi-format medical data.

The long-term healthcare impact of this type of work consists of two parts: to demonstrate general advantages of applying a specialized machine learning approach to healthcare data, and to argue for the use of multi-scale representation on complex medical data, integrated from multiple sources and containing rare events. Although the results presented here are not ideal (possibly due to complexity of the studied phenomena), they are sufficient to recommend the method for future use in healthcare predictive analytics.

Supporting Information

S1 File. This file contains the entire Supplementary Materials section within it. Table A, Comparative sensitivity results for ML(W)SVM against the regular SVM, WSVM, NB, C4.5, 5NN, and LR on Twonorm, Letter, Ringnorm, and Clean academic datasets for different fractions of missing values (r_{mv}) using the REM imputation method. **Table B,** Comparative specificity results for ML(W)SVM against the regular SVM, WSVM, NB, C4.5, 5NN, and LR on Twonorm, Letter, Ringnorm, and Clean academic datasets for different fractions of missing values (r_{mv}) using the REM imputation method. **Table C,** Comparative accuracy results for ML(W)SVM against the regular SVM, WSVM, NB, C4.5, 5NN, and LR on Twonorm, Letter, Ringnorm, and Clean academic datasets for different fractions of missing values (r_{mv}) using the REM imputation method. **Table D,** Computational time (sec.) for C4.5, 5NN, NB, LR, and MLSVM (excluding model selection) on public datasets. The results show that MLSVM is faster than other machine learning methods. In addition, we note that the average computational time of the REM imputation for public datasets over all missing value ratios are:

Twonorm 1.22, Letter 6.89, Ringnorm 1.18, cod-rna 33.76, Clean 7.85, Advertisement 0.57, Nursery 1.41, Hypothyroid 0.16, Buzz 1705.60 sec. respectively.
(PDF)

Author Contributions

Conceived and designed the experiments: TR, IS, OR, NM. Performed the experiments: TR, OR, IS. Analyzed the data: TR, OR, IS. Contributed reagents/materials/analysis tools: TR, IS, OR, NM. Wrote the paper: TR, IS, OR, NM.

References

1. Foldy S. National public health informatics, United States. In: Public Health Informatics and Information Systems. Springer; 2014. p. 573–601.

2. Haas LR, Takahashi PY, Shah ND, Stroebel RJ, Bernard ME, Finnie DM, et al. Risk-stratification methods for identifying patients for care coordination. The American journal of managed care. 2013; 19 (9):725–732. PMID: 24304255

3. Plis K, Bunescu R, Marling C, Shubrook J, Schwartz F. A Machine Learning Approach to Predicting Blood Glucose Levels for Diabetes Management. Modern Artificial Intelligence for Health Analytics Papers from the AAAI-14. 2014;.

4. Woolery LK, Grzymala-Busse J. Machine learning for an expert system to predict preterm birth risk. Journal of the American Medical Informatics Association. 1994; 1(6):439–446. doi: 10.1136/jamia. 1994.95153433 PMID: 7850569

5. Larson E, Bratts T, Zwanziger J, Stone P. A survey of IRB process in 68 US hospitals. Journal of Nursing Scholarship. 2004; 36(3):260–264. doi: 10.1111/j.1547-5069.2004.04047.x PMID: 15495496

6. Snomed, CT: Systematized Nomenclature of Medicine-Clinical Terms. International Health Terminology Standards Development Organisation. 2011;.

7. Roderick O. Why risk models and access to large data should work together: tangible advantages to merging clinical and claims data. Technical report, DARCI: Data Science, Geisinger Health System. 2014;.

8. Roderick O. Predictive model for response to medical outreach. Technical report, DARCI: Data Science, Geisinger Health System. 2015;.

9. Vapnik V. The nature of statistical learning theory. Springer; 2000.

10. Chang CC, Lin CJ. LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST). 2011; 2(3):27.

11. Graf HP, Cosatto E, Bottou L, Dourdanovic I, Vapnik V. Parallel support vector machines: The cascade SVM. In: Advances in neural information processing systems; 2004. p. 521–528.

12. Collobert R, Bengio S, Bengio Y. A parallel mixture of SVMs for very large scale problems. Neural Computation. 2002; 14(5):1105–1114. doi: 10.1162/089976602753633402 PMID: 11972909

13. Tveit A, Hetland ML. Multicategory incremental proximal support vector classifiers. In: Knowledge-Based Intelligent Information and Engineering Systems. Springer; 2003. p. 386–392.

14. Joachims T. Advances in Kernel Methods. MIT Press; 1999. p. 169–184.

15. Catak FO, Balaban ME. CloudSVM: training an SVM classifier in cloud computing systems. In: Pervasive Computing and the Networked World. Springer; 2013. p. 57–68.

16. Platt JC. Fast training of support vector machines using sequential minimal optimization. In: Advances in kernel methods. MIT press; 1999. p. 185–208.

17. Collobert R, Bengio S, Mariéthoz J. Torch: a modular machine learning software library. Technical Report IDIAP-RR 02-46, IDIAP; 2002.

18. Decoste D, Schölkopf B. Training invariant support vector machines. Machine Learning. 2002; 46(1–3):161–190. doi: 10.1023/A:1012454411458

19. Tang Y, Zhang YQ, Chawla NV, Krasser S. SVMs modeling for highly imbalanced classification. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on. 2009; 39(1):281–288. doi: 10. 1109/TSMCB.2008.2002909

20. Lo HY, Chang CM, Chiang TH, Hsiao CY, Huang A, Kuo TT, et al. Learning to improve area-under-FROC for imbalanced medical data classification using an ensemble method. ACM SIGKDD Explorations Newsletter. 2008; 10(2):43–46. doi: 10.1145/1540276.1540290

21. Mazurowski MA, Habas PA, Zurada JM, Lo JY, Baker JA, Tourassi GD. Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. Neural Networks. 2008; 21(2):427–436. doi: 10.1016/j.neunet.2007.12.031 PMID: 18272329

22. Li DC, Liu CW, Hu SC. A learning method for the class imbalance problem with medical data sets. Computers in Biology and Medicine. 2010; 40(5):509–518. doi: 10.1016/j.compbiomed.2010.03.005 PMID: 20347072

23. Kwak N. Feature extraction for classification problems and its application to face recognition. Pattern Recognition. 2008; 41(5):1701–1717. doi: 10.1016/j.patcog.2007.10.012

24. Batuwita R, Palade V. microPred: effective classification of pre-miRNAs for human miRNA gene prediction. Bioinformatics. 2009; 25(8):989–995. doi: 10.1093/bioinformatics/btp107 PMID: 19233894

25. Ezawa KJ, Singh M, Norton SW. Learning goal oriented Bayesian networks for telecommunications risk management. In: ICML; 1996. p. 139–147.

26. Groth SS, Muntermann J. An intraday market risk management approach based on textual analysis. Decision Support Systems. 2011; 50(4):680–691. doi: 10.1016/j.dss.2010.08.019

27. Su CT, Hsiao YH. An evaluation of the robustness of MTS for imbalanced data. Knowledge and Data Engineering, IEEE Transactions on. 2007; 19(10):1321–1332. doi: 10.1109/TKDE.2007.190623

28. Veropoulos K, Campbell C, Cristianini N, et al. Controlling the sensitivity of support vector machines. In: Proceedings of the International Joint Conference on Artificial Intelligence. vol. 1999; 1999. p. 55–60.

29. Lin CF, Wang SD. Fuzzy support vector machines. Neural Networks, IEEE Transactions on. 2002; 13 (2):464–471. doi: 10.1109/72.991432

30. Brandt A, Ron D. Chapter 1: Multigrid solvers and multilevel optimization strategies. In: Cong J, Shinnerl JR, editors. Multilevel Optimization and VLSICAD. Kluwer; 2003.

31. Razzaghi T, Safro I. Scalable Multilevel Support Vector Machines. In: International Conference On Computational Science. vol. 51. Procedia Computer Science; 2015. p. 2683–2687.

32. Muja M, Lowe DG. Scalable nearest neighbor algorithms for high dimensional data. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 2014; 36(11):2227–2240. doi: 10.1109/TPAMI.2014. 2321376

33. Muja M, Lowe DG. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. In: International Conference on Computer Vision Theory and Application VISSAPP'09). INSTICC Press; 2009. p. 331–340.

34. Sakellaridi S, ren Fang H, Saad Y. Graph-Based Multilevel Dimensionality Reduction with Applications to Eigenfaces and Latent Semantic Indexing. In: Machine Learning and Applications, 2008. ICMLA'08. Seventh International Conference on; 2008. p. 194–200.

35. Ron D, Safro I, Brandt A. Relaxation-Based Coarsening and Multiscale Graph Organization. Multiscale Modeling & Simulation. 2011; 9(1):407–423. doi: 10.1137/100791142

36. Safro I, Ron D, Brandt A. Graph minimum linear arrangement by multilevel weighted edge contractions. J Algorithms. 2006; 60(1):24–41. doi: 10.1016/j.jalgor.2004.10.004

37. Wang L. Feature selection with kernel class separability. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 2008; 30(9):1534–1546. doi: 10.1109/TPAMI.2007.70799

38. Huang CM, Lee YJ, Lin DKJ, Huang SY. Model selection for support vector machines via uniform design. Computational Statistics & Data Analysis. 2007; 52(1):335–346. doi: 10.1016/j.csda.2007.02. 013

39. Syed NA, Huan S, Kah L, Sung K. Incremental learning with support vector machines. In Proc of the Int Joint Conf on Artificial Intelligence (IJCAI). 1999;.

40. Fung G, Mangasarian OL. Incremental Support Vector Machine Classification. In: SDM. SIAM; 2002. p. 247–260.

41. Farhangfar A, Kurgan L, Dy J. Impact of imputation of missing values on classification error for discrete data. Pattern Recognition. 2008; 41(12):3692–3705. doi: 10.1016/j.patcog.2008.05.019

42. Ghannad-Rezaie M, Soltanian-Zadeh H, Ying H, Dong M. Selection–fusion approach for classification of datasets with missing values. Pattern recognition. 2010; 43(6):2340–2350. doi: 10.1016/j.patcog. 2009.12.003 PMID: 20212921

43. Little RJ, Rubin DB. Statistical analysis with missing data. 2002;.

44. Schafer JL. Analysis of incomplete multivariate data. CRC press; 2010.

45. García-Laencina PJ, Sancho-Gómez JL, Figueiras-Vidal AR. Pattern classification with missing data: a review. Neural Computing and Applications. 2010; 19(2):263–282. doi: 10.1007/s00521-009-0295-6

46. Gheyas IA, Smith LS. A neural network-based framework for the reconstruction of incomplete data sets. Neurocomputing. 2010; 73(16):3039–3065. doi: 10.1016/j.neucom.2010.06.021

47. Donders ART, van der Heijden GJ, Stijnen T, Moons KG. Review: a gentle introduction to imputation of missing values. Journal of clinical epidemiology. 2006; 59(10):1087–1091. doi: 10.1016/j.jclinepi.2006.01.014 PMID: 16980149

48. Batista GE, Monard MC. A Study of K-Nearest Neighbour as an Imputation Method. HIS. 2002; 87:251–260.

49. Oba S, Sato Ma, Takemasa I, Monden M, Matsubara Ki, Ishii S. A Bayesian missing value estimation method for gene expression profile data. Bioinformatics. 2003; 19(16):2088–2096. doi: 10.1093/bioinformatics/btg287 PMID: 14594714

50. Schneider T. Analysis of incomplete climate data: Estimation of mean values and covariance matrices and imputation of missing values. Journal of Climate. 2001; 14(5):853–871. doi: 10.1175/1520-0442(2001)014%3C0853:AOICDE%3E2.0.CO;2

51. Huang BC, Salleb-Aouissi A. Maximum entropy density estimation with incomplete presence-only data. In: International Conference on Artificial Intelligence and Statistics; 2009. p. 240–247.

52. Ghahramani Z, Jordan MI. Supervised learning from incomplete data via an EM approach. In: Advances in Neural Information Processing Systems 6. Citeseer; 1994.

53. Nanni L, Lumini A, Brahnam S. A classifier ensemble approach for the missing feature problem. Artificial intelligence in medicine. 2012; 55(1):37–50. doi: 10.1016/j.artmed.2011.11.006 PMID: 22188722

54. Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society Series B (Methodological). 1977;p. 1–38.

55. Redner RA, Walker HF. Mixture densities, maximum likelihood and the EM algorithm. SIAM review. 1984; 26(2):195–239. doi: 10.1137/1026034

56. Li H, Zhang K, Jiang T. The regularized EM algorithm. In: Proceedings of the national conference on artificial intelligence. vol. 20. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999; 2005. p. 807.

57. Razzaghi T, Safro I; 2016. Available from: http://www.cs.clemson.edu/~isafro/software.

58. Frank A, Asuncion A. UCI machine learning repository. [http://archive.ics.uci.edu/ml] irvine, ca: University of california, "School of Information and Computer Science". 2010;vol. 213.

59. Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, et al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proceedings of the National Academy of Sciences. 1999; 96(12):6745–6750. doi: 10.1073/pnas.96.12.6745

60. MATLAB. version 7.14.0.739 (R2012a). Natick, Massachusetts: The MathWorks Inc.; 2012.

61. Quinlan JR. Induction of decision trees. Machine learning. 1986; 1(1):81–106. doi: 10.1007/BF00116251

62. Russell SJ, Norvig P. Artificial intelligence: a modern approach (International Edition). 2002;.

63. Homser DW, Lemeshow S. Applied logistic regression. New York (NY): J Wiley and Sons. 1989;.

64. Cover TM, Hart PE. Nearest neighbor pattern classification. Information Theory, IEEE Transactions on. 1967; 13(1):21–27. doi: 10.1109/TIT.1967.1053964

65. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software: an update. ACM SIGKDD explorations newsletter. 2009; 11(1):10–18. doi: 10.1145/1656274.1656278

66. Mangasarian OL, Wild EW. Privacy-Preserving Classification of Horizontally Partitioned Data via Random Kernels. In: DMIN; 2008. p. 473–479.