Supplementary: e-Methods

Missing-indicator method

Missing data was a significant problem since the database originated from eight different hospitals and every hospital had their own completeness. Variables particularly affected (>5% missing data) were myocardial infarction (missing in 42.15%), dyslipidemia (missing in 52.97%), any alcohol current use (missing 29.32%), use of anticoagulants (missing 43.32%), use of anti-platelet (missing 33.71%), blood glucose (missing 14.22%), OTT (missing 5.10%) and DNT (missing 33.84%). We processed these characteristics using the missing-indicator method. A certain variable Q_i was introduced with respect to each real feature x_i, where Q_i referred to the existence of x_i. Then we replaced x_i with $Q_i x_i$, and inputted it along with $1 - Q_i$ into machine learning models. This process made the input feature matrix become sparse which to a certain extent would have negative impact on accuracy of our models. Therefore, features with a limited number of missing entries (<5%) were not filled using the missing-indicator method.

Implemental details of feature selection

Wrapper method, correlation-based feature selection in filter method and conservative mean feature selection were used to conduct the dimensional reduction process of the feature vector. Conservative mean method aims to maximize the correlation between features and label, compared with that of correlation-based feature selection. It calculated the AUC values between different features and the labels with K-fold validation methods. In this study, the wrapper method took the AUCs of the cross-validation result of a certain model as the corresponding merit function and chose the feature subset that could maximize it. Correlation-based feature selection method in filter method tended to maximize the correlation between feature subsets and label, meanwhile minimizing the correlation between different features. It can be expressed as equation (1), where $\overline{r_{cf}}$ denoted the averaged feature-class correlation, $\overline{r_{ff}}$ denoted the average feature-feature inter-correlation, k was the number of features in the feature subset, and $Merit_{S_k}$ denoted the heuristic "merit" of a feature

subset S containing k features.

$$Merit_{S_k} = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \quad (1)$$

For the correlation-based feature selection method, equal-frequency and equal-width discretization methods were used to better calculate the information entropy of features and label. Symmetrical uncertainty, RELIEF (relevant feature) and MDL (minimum description length) were set as the merit function to measure the merit of feature subsets.

To search for the optimal subsets in wrapper and filter method, forward search, backward search, and genetic algorithm and exhaustive search were used. When the dimension of feature sets was limited, exhaustive search was the best choice and could always find the global optimum. Yet when the dimension of feature sets was high, heuristic search algorithms were preferred, like forward and backward search and genetic algorithm. In our study, because of missing-indicator method, we manually restricted the states of $1 - Q_i$ and $Q_i x_i$ shared with each other because they were related to the same feature. For instance, if the factor "alcohol consumption" is true was put into the optimal feature subsets, the other factor concerning whether drinking is missing should be put into the subsets as well. This operation decreased the space to search from 2^{26} to 2^{18} in our case. Therefore, we chose the genetic algorithm, because exhaustive search was computationally expensive and forward and backward search easily converged to local optimum. In our study, Roulette Wheel selection and multiple-point crossover were used in the selection and crossover stages of the genetic algorithm. We then adapted the former merit function and set the fitness function of genetic algorithm to be (2). By setting power larger than one, difference between individuals will be expanded, which accelerated the process of convergence. The operations of deleting average from merit function and adding twice the difference between average and minimum to it was to make the power operation more effective. Otherwise, it was difficult for offspring to inherit the optimal features from their parents and converge to an optimum under the mechanism of Roulette Wheel selection and current merit function.

$$y_1 = Merit - average\ of\ group$$

$$y_2 = y_1 + 2 \times (average\ of\ group - minimum\ of\ group) \quad (2)$$

$$Fitness = (Merit)^{\gamma} \qquad \gamma > 1$$

Imbalanced data processing details

In this study, the over-sampling method was deployed, which sampled the minority examples until their total number approximated that of majority examples. During this process, the degree of class distribution balance was varied to be almost equal. Another measure that was taken was cost-sensitive adaptation. The loss function and learning rate were modified to penalize differently. For Logistic regression and perceptron, the loss function was cross-entropy. The weight that misclassified minority examples was higher than weights corresponding to other situations (3), was the cost function of original Logistic regression and perceptron, and (4) was the cost function after modification was made. $h_\theta\left(x^{(i)}\right)$ referred to the float value generated by machine learning algorithms. $y^{(i)}$ referred to the real label of a certain sample. Letter "m" was the number of samples in the data sets. The penalties of misclassifying positive samples and negative samples were determined by the value of $h_\theta\left(x^{(i)}\right)$ and the number of positive and negative samples. Then when using gradient descent to update parameters, $J(\theta)$ would be easily stuck in the local optimum where all the samples were labeled as majority class and in this case none-sICH type. Different weights ρ^+ and ρ^- were added to the cost function to punish various types of sample differently. sICH groups were taken as positive types and were multiplied by ρ^+, at the same time none-sICH groups were taken as negative types and were multiplied by ρ^-. ρ^+ was set to be larger than ρ^- in order to increase the punishment when the sICH samples were misclassified.

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} log h_\theta\left(x^{(i)}\right) + (1 - y^{(i)}) \log\left(1 - h_\theta\left(x^{(i)}\right)\right)] \quad (3)$$

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} \rho^+ y^{(i)} log h_\theta\left(x^{(i)}\right) + \rho^- (1 - y^{(i)}) \log\left(1 - h_\theta\left(x^{(i)}\right)\right)] \quad (4)$$

The learning rate in the gradient descent was adapted. For the samples belonging to majority class, we used a comparatively small learning rate. Meanwhile, for the samples belonging to minority class, we used a comparatively large learning rate, enabling the update of parameters to be more significant with regard to the minority class. It was demonstrated in the parameter updating equation (5), where cost function $J^{+}(\theta)$ merely contained sICH samples and $J^{-}(\theta)$ merely contained none-sICH samples.

$$\theta_j \coloneqq \theta_j - \alpha^{+}\frac{\partial}{\partial\theta_j}J^{+}(\theta) - \alpha^{-}\frac{\partial}{\partial\theta_j}J^{-}(\theta) \tag{5}$$

Multivariate SVM was also used to handle the problem that zero-one classification loss brought with imbalanced data. Setting AUC value as the loss function instead, the SVM turned out to have better performance. The original SVM problem after modification could be presented as following. Here $\Delta(\bar{y}', \bar{y})$ denoted non-linear AUC value instead of linear zero/one error rate. This optimization problem and its restriction could be expressed in (6) and (7).

$$min_{\boldsymbol{w},\xi\geq 0}\frac{1}{2}\boldsymbol{w}\cdot\boldsymbol{w} + C\xi \tag{6}$$

$$\text{s.t.: } \forall\bar{y}' \in \bar{\mathcal{Y}}\backslash\bar{y}\text{: } \boldsymbol{w}^T[\Psi(\bar{\boldsymbol{x}}, \bar{y}) - \Psi(\bar{\boldsymbol{x}}, \bar{y}')] \geq \Delta(\bar{y}', \bar{y}) - \xi \tag{7}$$

Explicit supervised ranking strategy

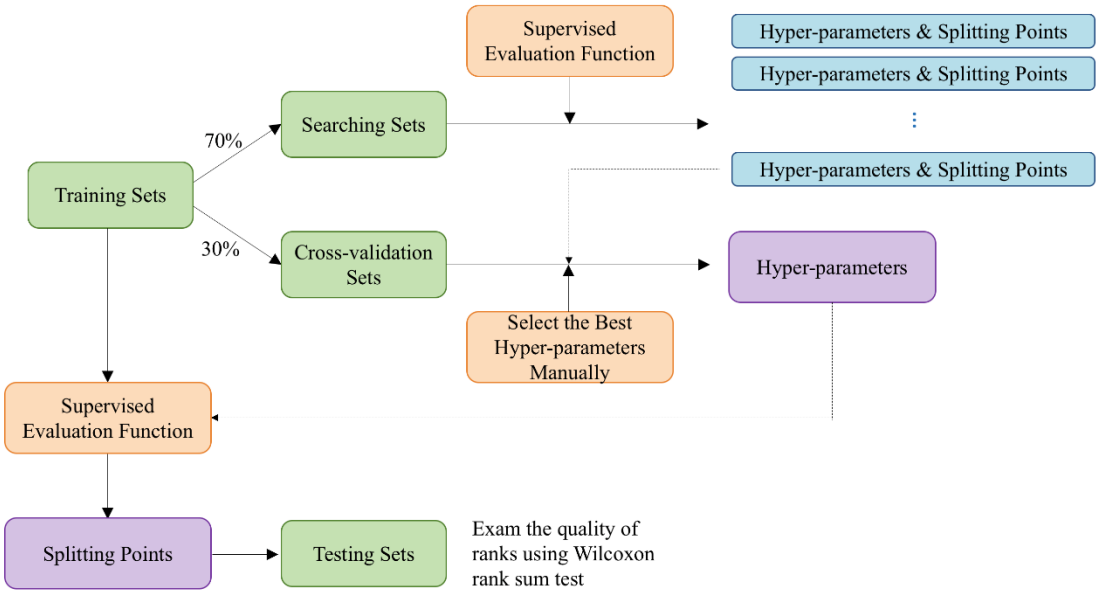
The supervised method was mainly established on the principle of minimizing entropy. However, the splitting points set which could minimize entropy were not necessary the ones best suited for clinical use. For instance, the practice of minimizing information entropy would elicit problem like selecting an extremely narrow range which could minimize the overall information entropy yet violate the medical requirement for discretization. Therefore, we added two penalty terms to the overall information entropy formula to construct an evaluation function serving to find the set of splitting points with which our needs could be satisfied. In this case, we could manually set the approximate expected width of each rank, and a special case for this is the former unsupervised equal frequency method. We set the first penalty term to

measure the extent to which the widths of calculated splitting points deviated the expected widths. It could be suggested by the equation (8). n refers to the number of ranks; L refers to the total number of training sets; $\widetilde{W_i}$ refers to the number of samples in the i^{th} rank which is generated by the algorithm; W_i refers to the expected number of samples in the i^{th} rank.

$$Penalty_1 = \sum_{i=1}^{n} (\frac{\widetilde{W_i}}{L} - \frac{W_i}{L})^2 \quad (8)$$

From the perspective of clinical practice, we increased the sICH rate in the highest rank to the overall number of the highest rank. The accuracy of estimation in this group was considered more important than that in other groups. We set the second penalty term to be the inverse of the aforementioned rate. In sum, the evaluation function for the supervised discretization methods consisted of information entropy and two penalty terms, which were multiplied by corresponding weights to balance their influence on the choice of splitting points. These weights were hyper-parameters to be determined by cross-validation.

We first partitioned training sets into searching sets and cross-validation sets, which occupied 70% and 30% of the original training sets respectively. Then a group of possible hyper-parameters was input into the supervised discretization method, and a number of results were proposed by minimizing the aforementioned evaluation function. Exhaustive search for optimal splitting-point set was executed over searching sets, and results were produced with respect to cross-validation sets. We selected the results that best satisfy our expectation and their corresponding hyper-parameters. Next, supervised discretization method was conducted over the whole training sets under the hyper-parameters we got from the former step, and splitting points were generated.



Code implementation of the system

The code implement of this system was based on the Python 3.6 platform. The results of Fisher's exact test and Kruskal-Wallis test were figured out with 'SciPy 0.19.1'. The implementation of feature selection was based on the 'numpy 1.11.3' package. As for classifiers, the multivariate SVM was calculated using 'svm-perf' package in C and the original SVM was calculated using 'libsvm 3.22' package. Random forest was established by using 'Scikit-Learn 0.18.1' package. For Logistic regression and neural network, 'Theano 0.9.0' and 'TensorFlow 1.2.1' were both used.

The overall schematic of the machine learning process

(A) Overview of the machine learning system: It demonstrates the structure of the machine learning systems for the prediction of sICH after stroke thrombolysis. The whole dataset of AIS first went through data preprocessing which included missing data imputation, data cleansing and normalization. Then, the training set went through feature selection, imbalanced processing, classifier and discretization. This process generated parameters and the testing set was used to evaluate the accuracy of this system. (B) Structure for wrapper and filter feature selection: wrapper and filter feature selection provided different merit functions to evaluation process, yet they share the same searching process. (C) Structure for ranking: ranking was conducted regarding the training set. First, a search was performed to find optimal

hyper-parameters; then hyper-parameters were used to figure out split point groups.