Recent Developments in the Machine Learning Techniques for Survival Analysis

Yukyoung Seo : Master degree candidate, Department of Statistics, Pusan National University (uiop924@pusan.ac.kr)

Choongrak Kim : Professor, Department of Statistics, Pusan National University

Introduction

- Survival analysis is interested in the survival time, which is usually censored, until an event of interest occurs, and is currently applied to various fields such as medicine, biology, engineering, sociology, and economics.
- The Cox proportional hazard model (Cox, 1972) is a widely used model due to its analytical advantage because it expresses the relationship between the response variable and the covariates in the form of a risk function. However, in high-dimensional data such as genomic data, it is difficult to accurately predict the survival time because a ill-conditioned matrix is generated due to the correlation between variables.
- To solve this problem and improve the performance in predicting survival time, the need for a model that explains the nonlinear relationship or interaction between covariates has been raised. Therefore, a machine learning technique that facilitates high-dimensional censored data is required a lot in survival analysis.

Method

- **Cox proportional hazard model (Cox,1972)**

$$h(t\colon \boldsymbol{x}) = h_0(t)\exp(\boldsymbol{x}'\beta)$$

- $h(t\colon \boldsymbol{x})$ is the risk rate for individuals with a risk factor or covariates $\boldsymbol{x}$ at time t.
- To estimate the regression coefficient β, using partial likelihood(Cox, 1975)
- Negative log-partial likelihood function:

$$l(\beta) = -\sum_{i=1}^{n}\delta_i\{x'_i\,\beta - \log[\sum_{j\in R(t_i)}\exp(x'_j\,\beta)]\}$$

- **Kernel Cox(Li and Luan, 2002)**
- To overcome the problem of high-dimensional data, it is solved by giving a penalty for complexity in Cox regression.

$$R(f) = -\frac{1}{n}\sum_{i=1}^{n}\delta_i\{f(x_i) - \log[\sum_{j\in R(t_i)}\exp(f(x_i))]\} + \xi\|f\|_{H_K}^2\ ,where\ f(x) = b + \sum_{i=1}^{n}a_iK(x,x_i)$$

- **Regularization Cox**
- In high-dimensional data such as genomic data, it is difficult to accurately predict the survival time because a ill-conditioned matrix is generated due to the correlation between variables.
- Assuming that only a few of the many genes affect disease, select significant variables and shrink the regression coefficients.
- Types of regularization : Lasso Cox (Tibshirani, 1997), Ridge Cox(Verweij and Van Houwelingen, 1994), Elastic-net Cox(Simon et al., 2011)

$$\hat{\beta}_L = argmin\{-\sum_{i=1}^{n}\delta_i\{x'_i\,\beta - \log[\sum_{j\in R(t_i)}\exp(x'_j\,\beta)]\} + \lambda\sum_{k=1}^{p}|\beta_k|\}$$

$$\hat{\beta}_R = argmin\{-\sum_{i=1}^{n}\delta_i\{x'_i\,\beta - \log[\sum_{j\in R(t_i)}\exp(x'_j\,\beta)]\} + \lambda\sum_{k=1}^{p}{\beta_k}^2\}$$

$$\hat{\beta}_{EN} = argmin\{-\sum_{i=1}^{n}\delta_i\{x'_i\,\beta - \log[\sum_{j\in R(t_i)}\exp(x'_j\,\beta)]\} + \lambda\sum_{k=1}^{p}|\beta_k| + (1-\lambda)\sum_{k=1}^{p}{\beta_k}^2\}$$

- **Machine learning techniques for survival analysis**
- A machine learning technique that facilitates high-dimensional censored data is required a lot in survival analysis.

① Random survival forest

- Random Forest is an ensemble method that uses a survival tree as the base learner for prediction.
- RSF is a special case of bagging, in the bootstrap process, unlike bagging, which uses all of the features, only some features are selected and splitted.
- The prediction performance can be improved by reducing the correlation between trees through randomization.
- In RSF, the concern is which features make up a subset of features when nodes are splitted. Here, when randomly selecting a variable, we use the variable importance(VIMP), which is a measure of the increase or decrease in the prediction error for the forest ensemble. In addition, the permutation importance (Altmann et al., 2010)

② gradient survival boosting

- The predictions are combined in an additional way, adding each base learner improves the overall model.
- Let $L(y, f(X))$ call as the loss function. Where $f(X)$ is the statistical model. The goal is to estimate $f(X)$ by iteratively updating the values through the base learner $g(y, X)$.

```
Algorithm 1 Gradient boosting algorithm
Step 1. Initialize the estimate.
Step 2. Compute the negative gradient vector, $u = -\frac{\partial L(y,f(X))}{\partial f(X)}\Big|_{f(X)=\hat{f}(X)}$
Step 3.
1) Fit the base learner to the negative gradient vector, $\hat{g}(u, X)$
2) Penalize the value, $\hat{h}(X) = \nu\hat{g}(u, X)$
Step 4. Update the estimate, $\hat{f}(X) = \hat{f}(X) + \hat{h}(X)$
```

- Steps 2-4 are repeated in the algorithm for m, which is the number of boosting iterations.
- The loss function used in the boosting algorithm is the partial likelihood loss of Cox's proportional hazard model.

③ Component-wise gradient survival boosting

- To cope with high-dimensional data, component-wise boosting was applied in survival analysis
- A component-wise boosting, with the algorithm described above modified to update $f(x)$ using only one dimension of X, at each boosting iteration.

```
Algorithm 2 Component-wise gradient boosting algorithm
Step 1. Initialize the estimate. e.g., $\hat{\beta} = (0, \dots, 0)$;
Step 2. Compute the negative gradient vector, $u = -\frac{\partial L(y,f(X,\beta))}{\partial f(X,\beta)}\Big|_{\beta=\dot{\beta}}$;
Step 3.
1) Fit the base learner to the negative gradient vector, $\hat{g}(u, X_j)$
2) Penalize the value, $\hat{b}_j = \nu\hat{g}(u, X_j)$
Step 4. Select the best update $j^*$ that minimizing the loss function
Step 5. update the estimate, $\hat{\beta}_{j^*} = \hat{\beta}_{j^*} + \hat{b}_j$
```

- Repeat m times from 2 to 5. For component-wise boosting using least squares as the base learner, the final model will be a linear model.

④ Survival support vector machine

- The basic idea of a support vector machine is to find a hyperplane that maximizes the margin between the two classes while minimizing error.
- This was extended and applied to survival analysis problems: support vector regression for censored data (SVRc)
- Rank SVM is a method of assigning samples with shorter survival times to a lower rank,

Taking into account all possible pairs of samples P in the data.

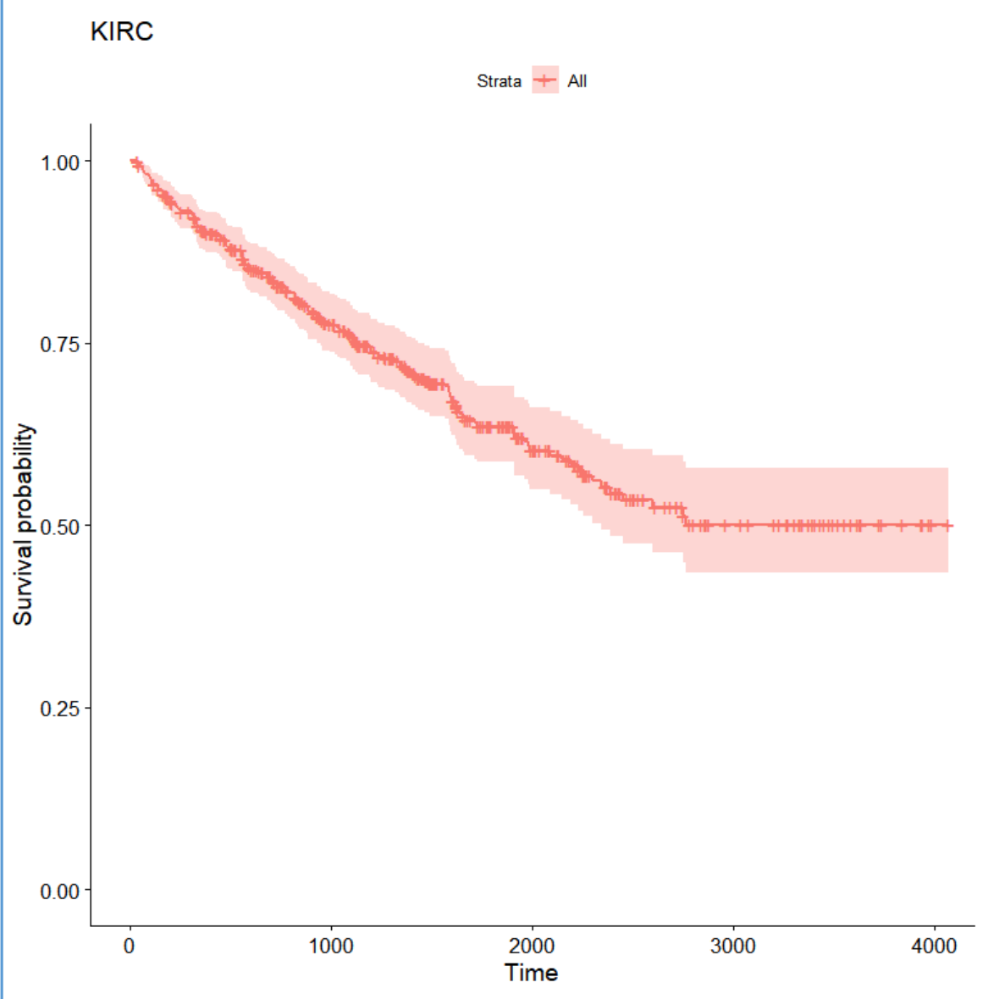
$$P = \{(\mathrm{i},\mathrm{j})|\ y_i >\ y_j\ s.t\ \delta_j = 1\}, where, i, j = 1, \dots, n$$

- The objective function $f(w)$ to minimize is

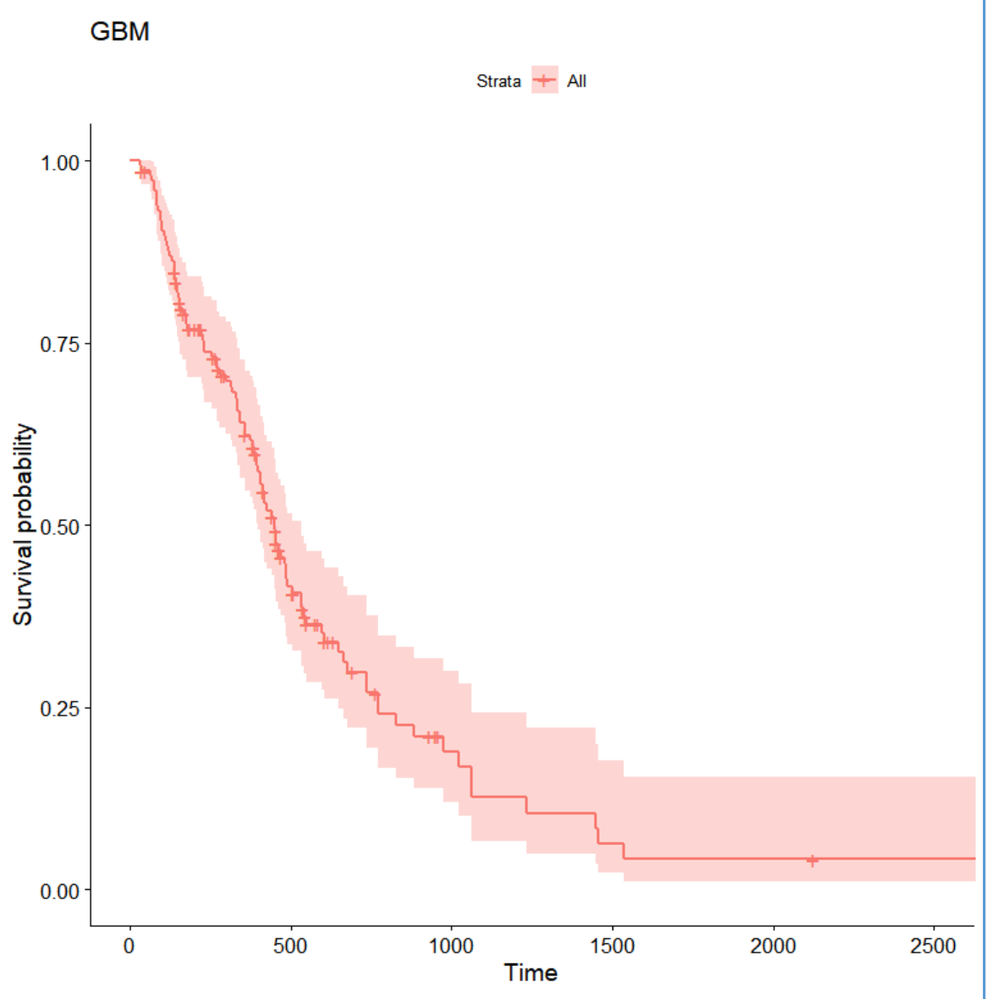
$$f(w) = \frac{1}{2}\|\beta\|_2^2 + \frac{\gamma}{2}\sum_{i,j\in P}\max(0, 1-\beta'(x_i - x_j))^2$$

Data analysis

- Well-known machine learning techniques such as random forest, boosting, and support vector machines are introduced and applied to both KIRC (Kidney Renal Clear Cell Carcinoma) and GBM(Glioblastoma multiforme) clinical data with RNA-sequencing information.
- Each of the two data has a different censoring percentage: heavy censoring and light censoring.
- The performance of several machine learning techniques are compared using the concordance index (Harrell et al.,1982).



Figure(a)



Figure(b)

- GBM data is light censored data with a censoring percentage of 32.4%, and KIRC data is heavy censored data with a censoring percent of 67.3%
- In Figure(a), it has a relatively constant slope up to 2,500 days.
- In Figure(b), as indicated by the steep slope of the estimated survival function over the first 500 days, it can be seen that most of the patients died within the first 500 days.

Concordance index

- Before applying the above machine learning techniques to real data, consider the performance indicators of survival analysis.
- When trying to assess the prediction performance of a survival analysis model, metrics such as the mean squared error are not suitable due to the presence of censored data.
- Compare the above analysis results with a concordance index(Harrell et al.
- Concordance index (where, y_i:observed value, $\hat{y}_i$:predicted value,δ_j: censoring indicator):

$$c = \frac{I(y_i > y_j)I(\hat{y}_i > \hat{y}_j)\delta_j}{I(y_i > y_j)\delta_j},\ \ 0 \le c\ \le 1$$

- If c is closer to 1, it is interpreted that it predicts better, and if it is closer to 0.5, it is evaluated as predicting randomly.

Data preprocessing

- For this analysis, among many tools, python package “scikit-survival” is used. These data contain RNA-seq data and clinical information.
- Genes with zero expression in more than 50% of the samples were removed and the genes were normalized using R package “limma”.
- There are three main methods of data analysis(Lee and Lim, 2019):
  1) Only clinical data
  2) Only significant gene data
  3) both 1) and 2)
- Using regularization to extract some significant genes. → lasso and elastic-net are used.
- The process of selecting a significant gene is as follows:
  1) After selecting 300 important genes using lasso regularization
  2) 5-fold cross-validation (3-fold cross-validation in GBM data) selects the number of genes at the optimal hyper parameter (i.e., the highest c-index).
- In (3), the clinical variables are selected through lasso regularization and elastic-net regularization by 3-fold cross validation. Then compare the results of these.
- The hyper parameters of all models are set as default values.
- c-index is calculated as follows:
  1) Calculate the average c-index using a 5-fold (or 3-fold) cross validation.
  2) Repeat this 100 times to average 100 c-indexes to calculate the final c-index.

Data description

① KIRC data

- There were 171 genes selected under Lasso regularization and 216 genes selected under elastic-net regularization.
- Also, the number of selected clinical variables is 8 and 10, respectively.

② GBM data

- 101 genes were selected both under Lasso normalization and under elastic net regularization. The number of genes is the same, but the type of gene selected is different.
- In both, 8 clinic variables were selected.
- Cox proportional hazard model was unable to calculate the c-index because it was impossible to estimate the coefficients due to ill-conditioned matrix.

conclusion

- As a result of data analysis, in the case of KIRC data, the best results were obtained when only clinical data were used. Also, it was found that the performance of the survival support vector machine was generally good.
- Meanwhile, GBM data performed best when both clinical and genetic data were used. In addition, it was found that the results of using the elastic net gene were better than that of the Lasso gene.

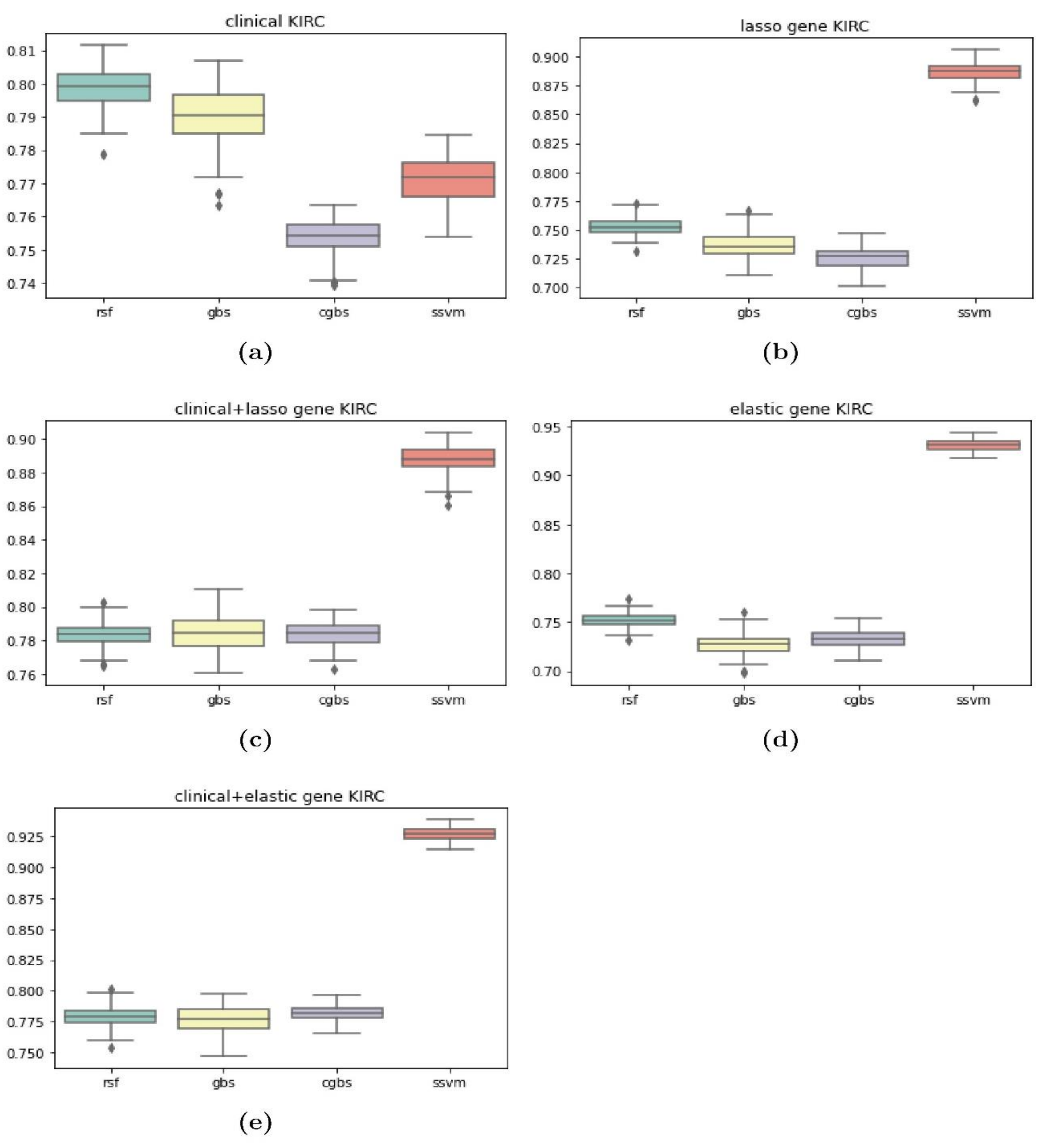
Average of 100 c-index of KIRC data

KIRC data		Random survival forest	Survival gradient boosting	Componentwise Survival gradient boosting	Survival support vector machine
clinical		0.799	0.79	0.754	0.771
lasso	Gene	0.753	0.737	0.726	0.887
	gene+ clinical	0.784	0.784	0.784	0.888
Elastic net	Gene	0.752	0.728	0.733	0.931
	Gene+ clinical	0.779	0.777	0.782	0.927

Average of 100 c-index of GBM data

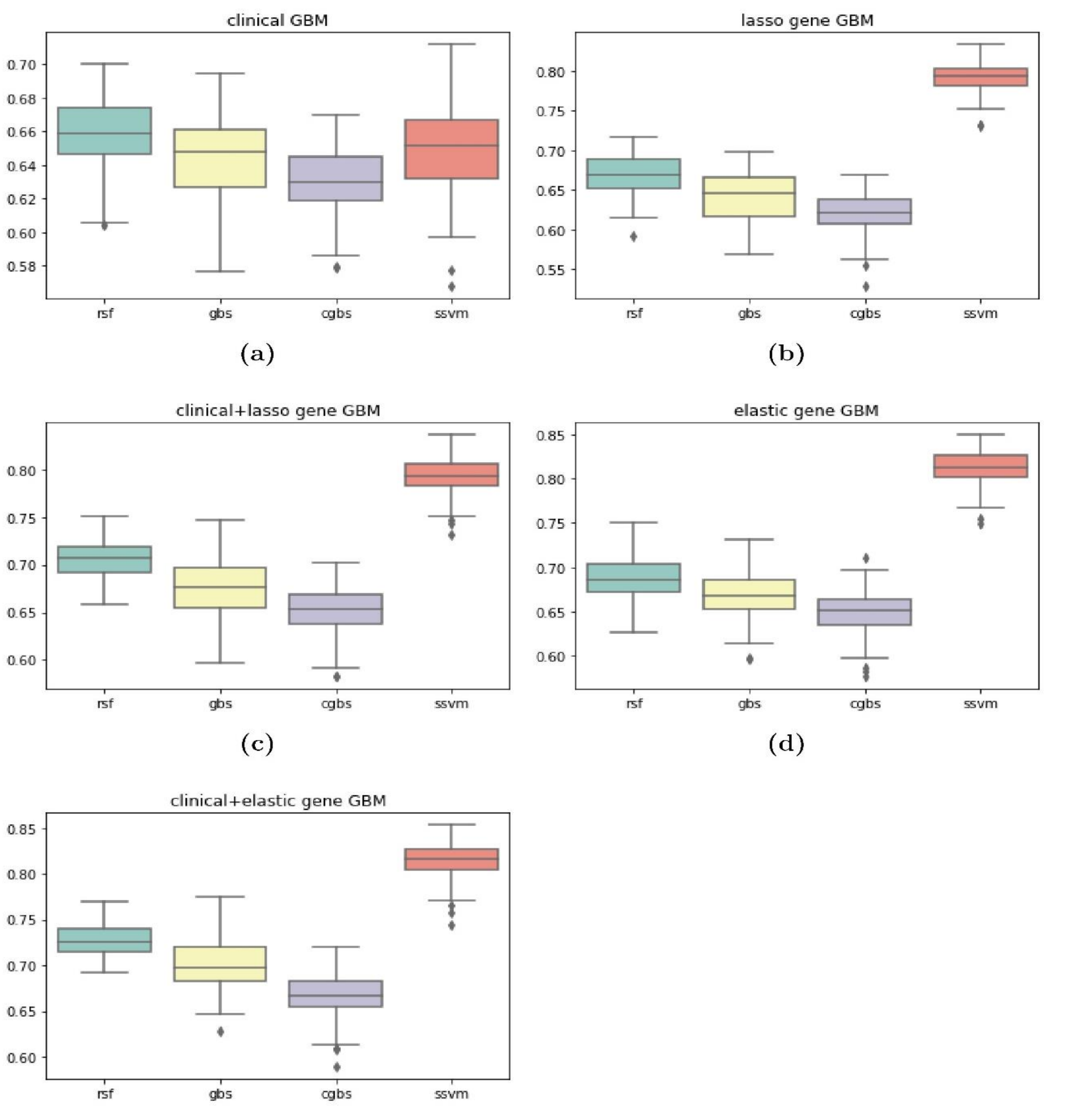
GBM data		Random survival forest	Survival gradient boosting	Componentwise Survival gradient boosting	Survival support vector machine
clinical		0.659	0.644	0.629	0.650
lasso	Gene	0.670	0.641	0.620	0.792
	gene+ clinical	0.706	0.675	0.652	0.792
Elastic net	Gene	0.688	0.688	0.649	0.812
	Gene+ clinical	0.727	0.698	0.667	0.815

Result of KIRC data



(a) (b) (c) (d) (e)

Result of GBM data



(a) (b) (c) (d) (e)