2. Git и GitHub: что это такое и в чём разница [Электронный ресурс]. URL: https://tproger.ru/translations/difference-between-git-and-github/ (дата обновления: 10.03.2020).

3. Взаимодействие между Javascript и CSS с помощью CSS-переменных [Электронный ресурс]. URL: https://css-live.ru/articles/vzaimodejstvie-mezhdu-javascript-i-css-s-pomoshhyu-css-peremennyx.html (дата обновления: 10.03.2020).

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ПОСТРОЕНИЯ МНОЖЕСТВ МОДЕЛЕЙ ВЫЖИВАЕМОСТИ НА ВЫБОРКАХ ОГРАНИЧЕННОГО ОБЪЁМА В МЕДИЦИНСКИХ ИССЛЕДОВАНИЯХ

SOFTWARE FOR BUILDING OF SETS OF SURVIVAL MODELS ON SAMPLES OF LIMITED SIZE IN MEDICINE RESEARCHES

А. В. Копыцкий, В. Н. Хильманович, Т. Н. Сакович, А. К. Пашко, В. М. Завадская

A. Kapytski, V. Khilmanovich, T. Sakovich, A. Pashko, V. Zavadskaya

Гродненский государственный медицинский университет,
г. Гродно, Республика Беларусь
fizika@grsmu.by
Grodno State Medical University, Grodno, Republic of Belarus

Анализ выживаемости – один из важных аспектов современной медицинской науки. Однако, зачастую объёмы выборок в медицинских исследованиях невелики, а число показателей, измеренных у пациентов, наоборот, велико. В таких случаях, построение моделей выживаемости при одновременном включении или при пошаговом включении (исключении) переменных-предикторов оказывается невозможным. Кроме этого, многие показатели состояния здоровья пациента могут быть статистически значимо связаны, что приводит к проблеме мультиколлинеарности предикторов в модели. Одновременное решение обеих проблем: малого объёма и мультиколлинеарности – прямой перебор множеств моделей выживаемости, построенных на всех сочетаниях предикторов из определённого подмножества. Нами разработано программное решение на языке «R», позволяющее проводить перебор моделей выживаемости, построенных на отфильтрованных сочетаниях предикторов из их некоторого подмножества.

Survival analysis is one of the important aspects of modern medical science. However, often sample sizes in medical researches are small, but the number of health indicator measured in patients vice versa is high. In such cases, the construction of survival models with the total inclusion of predictors or forward (backward) stepwise methods is impossible. Moreover, many of patients' health indicators can be statistically significantly related, what leads to the problem of multicollinearity of model predictors. The simultaneous solution of both problems: small sample size and multicollinearity is a direct enumeration of the sets of survival models built on all combinations of predictors from a certain subset. We developed a software solution based on the "R" programming language, which allows enumerating survival models built on filtered combinations of predictors from a certain subset of them.

Ключевые слова: программное обеспечение, медицинские исследования, выборки малого объёма, модели выживаемости.

Keywords: software, medicine researches, small samples sizes, survival models.

https://doi.org/10.46646/SAKH-2020-2-315-318

Одной из актуальных проблем современной медицинской науки является проблема прогнозирования сроков наступления какого-либо события: смерти пациента, наступления рецидива заболевания, возобновления роста опухоли и т.д. На сегодняшний день существует несколько вариантов построения прогнозов. Среди них можно выделить две группы методов: методы машинного обучения и методы моделирования. Из методов машинного обучения обычно используются: метод опорных векторов, случайный лес, бустинг, нейронные сети [1]. Однако, наряду с более высокой (по сравнению с методами моделирования) точностью предсказания длительности жизни, эти методы отличаются худшей содержательной интерпретацией; используя данные методы, исследователь часто не может логически объяснить влияние факторов и ковариат на зависимую переменную. Кроме этого,

традиционные методы анализа предикторов выживаемости (например, модель пропорциональных рисков Кокса) уже широко распространены и часто используются как общепринятые стандартные процедуры.

Кроме того, одной из распространённых проблем, связанных с медицинскими данными, является проблема наличия пропущенных значений показателей, что приводит к тому, что количество полных строк данных пациентов часто оказывается мало для построения моделей выживаемости (модели Кокса или обобщённой аддитивной модели выживаемости), где требуется как минимум 10 измерений на один предиктор. В некоторых работах озвучивается и большее число: >20 [2], хотя в ряде работ, справедливо указывается, что при определённых условиях (например, при наличии априорных предположений о влиянии предикторов на выживаемость) это требование может быть несколько смягчено или наоборот усилено. При малых объёмах выборок также затруднительно использовать методы пошагового включения или исключения предикторов с целью определения наилучшего подмножества предикторов. Одним из наиболее простых решений, позволяющих одновременно решить проблему пропущенных значений, малого объёма выборки и получения наилучшего множества предикторов, является прямой перебор моделей выживаемости. Таким образом, актуальной является задача построения и анализа множества моделей выживаемости на выборках ограниченного объёма.

Решение данной задачи сопряжено с рядом трудностей. Во-первых, прямой перебор моделей для всех возможных сочетаний предикторов (без учёта их взаимодействия) из некоторого подмножества является довольно затратной по количеству вычислений процедурой. Во-вторых, обычно в медицинских исследованиях количество возможных показателей довольно велико: показатели общего и биохимического анализа крови, данные генетических исследований, данные ультразвуковой диагностики, симптомы заболеваний, данные анамнеза, заболевания родственников, гистологические параметры и т.д. Количество комбинаций, которые можно составить из десятков показателей может быть огромным, и добавление нового предиктора в модель увеличивает кратно количество вычислений. В целом, рост числа предикторов приводит к экспоненциальному росту числа операций. Однако, перечисленные трудности могут быть нивелированы рядом следующих соображений.

• Модель выживаемости не должна быть слишком сложной. По нашему опыту построения и изучения моделей выживаемости на реальных медицинских данных число предикторов в них редко превосходит 10. Например, в последних нескольких опубликованных на портале «PubMed» работах, где упоминается модель Кокса, максимальное число предикторов не превосходило 7. Простота модели обеспечивается небольшим количеством предикторов и возможностью их логичной интерпретации; кроме того, в оценки качества подгонки модели вводятся штрафы за каждый включённый предиктор, так что сами методы оценки качества натурально отдают предпочтение моделям с меньшим числом предикторов. Таким образом, при переборе подмножеств предикторов можно ограничиваться их небольшим количеством, что позволяет существенно уменьшить число возможных комбинаций и уменьшить время перебора.

• При построении моделей существенной является проблема мультиколлинеарности предикторов. Матрица модели будет т.н. плохо обусловленной, что приведёт к получению неустойчивых оценок регрессионных коэффициентов, и на новых данных модель будет иметь плохую предсказательную способность. Обычно перед построением регрессионных моделей и моделей выживаемости проводят корреляционный анализ для нахождения пар связанных (коррелирующих либо ассоциированных) предикторов с целью исключить из модели коллинеарные независимые переменные. В таком случае исследователь сам определяет, какой предиктор из пары для него важнее, и оставляет его для последующего анализа. Но при большом числе сложно связанных переменных всегда есть риск того, что отброшенная переменная была бы лучшим предиктором, чем оставленная. В таком случае исследователю необходимо повторно проверить модель, но уже при смене предикторов. Одновременным решением проблемы мультиколлинеарности и большого числа комбинаций предикторов являются предварительный автоматизированный анализ связей между переменными и последующая фильтрация списка сочетаний предикторов. При фильтрации списка из него нужно исключить комбинации, в которых встречается пара (или пары) связанных переменных. Таким образом, с одной стороны список комбинаций будет прорежен (и уменьшено время перебора), с другой стороны, модели, построенные на оставшихся сочетаниях предикторов, будут свободны от проблемы мультиколлинеарности.

Также для уменьшения времени расчётов можно использовать интенсивные методы обработки информации:

• Использовать параллелизацию вычислений. Для современных многоядерных компьютеров задача перебора моделей легко распараллеливается, так как каждая модель строится независимо от других.

• Использовать для построения моделей специализированные библиотеки, написанные на языках общего назначения (например библиотеку «Armadillo», реализованную на языке «C++» [3]), а подготовку данных и их окончательное представление (осуществляемые в начале перебора и в конце) можно выполнять на более высоком уровне, например с использование языка «R» [4]. Нам видится перспективным использование уже хорошо себя зарекомендовавшей связки библиотеки «Armadillo» (специализированной на векторной алгебре) и языка «R» (специализированного на статистическом анализе) посредством пакета расширения «RcppArmadillo» [5].

Исходя из вышеизложенного, нами подготовлено программное решение для прямого перебора множества моделей выживаемости для выборок ограниченного объёма. Программное решение выполнено на основе языка «R» с использованием пакета расширения «survival».

Программное решение имеет следующую архитектуру:

Рисунок 1 – Архитектура программного решения прямого перебора моделей выживаемости

Как видно из схемы (см. рисунок), в решении можно выделить 5 модулей:

• Модуль 1 – «ввод данных». В этом модуле обеспечивается ввод данных пользователя, подразумевается, что это будет электронная структурированная таблица, состоящая из m столбцов (из которых максимальное число предикторов m-3) и n строк (наблюдений). Тут же осуществляется разметка данных – определяется их тип: числовые или факторные переменные.

• Модуль 2 – «спецификация модели». В этом модуле пользователь указывает параметры модели – указывает индексы зависимых переменных: события, времени до события, цензурирующей переменной; указывает, какие переменные будут использоваться как потенциальные предикторы модели; выбирает тип модели: модель Кокса или аддитивная модель выживаемости; здесь же накладываются ограничения на модели.

• Модуль 3 – «построение множеств сочетаний индексов и их фильтрация». В этом модуле перебираются все возможные сочетания из k предикторов отобранных из множества m потенциальных предикторов, выбранных в модуле 2. При необходимости пользователь может сузить полученный набор матриц, указав в модуле 2, какие предикторы обязательно должны входить в итоговый набор. Здесь же проводится анализ связей показателей, определяются пары связанных переменных. После этого множество сочетаний предикторов фильтруется.

Для определения пар связанных предикторов нами разработано программное решение на языке «R», которое позволяет определять наличие или отсутствие связей между переменными. В данном решении сначала определяется тип пары предикторов. Возможны следующие пары: «фактор-фактор», «ковариата-фактор», «ковариата-ковариата». Для пары «фактор-фактор» строится таблица сопряжённости, для которой индикатором связи выступает χ^2-статистика или статистика точного теста Фишера. Для определения наличия связи в паре «ковариата-фактор» могут быть использованы как параметрические подходы: критерий Уэлча или однофакторный дисперсионный анализ, либо непараметрические критерии: Манна – Уитни или Краскела – Уоллиса. Для пары «ковариата-ковариата» мерами связи могут быть значения коэффициентов корреляции: Пирсона, Спирмена или Кендалла. Также вместо статистик критериев, проверяющих гипотезы об отсутствии связи между переменными в генеральной совокупности, можно использовать так называемые размеры эффектов и их максимальное значение, превышение которого можно рассматривать как наличие связи. Для нашего модуля это: бисериальный и рангово-бисериальный коэффициенты корреляции, коэффициент детерминации, коэффициент ε^2, сами коэффициенты корреляции.

• Модуль 4 – «построение множеств моделей и их фильтрация». По сути, это ядро программы, где анализируются все модели (специфицированные в модуле 2) с индексами предикторов, полученными в модуле 3. Здесь же при необходимости происходит фильтрация моделей, из которых оставляются только те, что удовлетворяют критериям, описанным в модуле 2.

• Модуль 5 – «вывод результатов перебора моделей». По окончании работы модуля 4 результаты перебора выводятся пользователю в этом модуле. На текущий момент характеристики полученных моделей выводятся в итоговый файл в виде электронной таблицы (формата «xlsx»). В этой же таблице на отдельных вкладках сохраняются параметры модели и исходная таблица данных.

После построения архитектуры была написана программа-прототип. Программа на текущий момент состоит из трёх файлов:

• «main.r» – где реализованы модули 1 и 2. Здесь пользователь подключает базу данных, структурирует данные. Для получения базы данных используется либо стандартная библиотека «utils»(для чтения «csv» файлов), либо библиотека «openxlsx» (для чтения «xlsx» файлов). Далее пользователь задаёт свойства модели: указывает индексы зависимых переменных, индексы предикторов, задаёт веса модели: либо вручную, либо использует функцию «get.weights» (см. далее).

• «core.r» – здесь реализованы модули 3, 4, 5. Производится построение множеств возможных сочетаний индексов предикторов, и производится перебор моделей, фиксируется время, затраченное на перебор из заданного подмножества, прогнозируется время окончания процедуры полного перебора. После окончания перебора результаты сохраняются в виде электронной таблицы формата «xlsx».

• «functions.r» – вспомогательный файл, содержащий функции, необходимые для работы всех модулей. Самая ёмкая часть программы. Содержит, в том числе, следующие важные функции:

o «get.formula(x,data.title)» – используется для получения модельной формулы в общепринятом виде: «Surv(time = …, event = …) ~ v1 + v2 + ….». В данной функции, «x» – вектор с индексами зависимых переменных и предикторов, «data.title» – название таблицы с данными, по которым строится модель. «Surv» – модель выживаемости, «time» – имя переменной-времени, «event» – имя переменной-события, «v1», «v2»,…– имена предикторов, включённых в модель;

- «get.model.info (x, par=…)» – центральная функция программы. Проводит построение и анализ модели, формула которой определяется ранее описанной функцией «get.formula(x,data.title)»;
- «get.weights(response.index, df)» – функция для автоматического получения весов наблюдений. Аргументы: «response.index» – индекс переменной-отклика (чаще всего индекс переменной-события), «df» – таблица данных, в которой находится данная переменная. Функция используется для балансировки данных, придавая дополнительный вес наблюдениям, встречающимся реже;
- «get.long.formula(model)» – функция для получения видоизменённой формулы модели, где учитывается знак регрессионного коэффициента предиктора и его полное название, что позволяет при анализе результатов перебора пользователем видеть, как каждая переменная-предиктор влияет на отклик.

Результат работы программы – электронная таблица с информацией об отобранных по результатам перебора и фильтрации моделям. Ниже (см. таблицу) приведён пример данной таблицы. В примере использованы следующие обозначения: «formula» – формула модели; «aic» – значение информационного критерия Акаике; «aic0» – значение информационного критерия Акаике для нуль-модели; «R2_MF» – псевдо R2-Мак Фаддена; «R2_Nag» – псевдо R2-Нагелькерке; «conc» – коэффициент конкордации; «χ2» – статистика χ2-отношения правдоподобия; «df» – число степеней свобода для χ2-статистики; «p.value» – p-значение для отношения правдоподобия; «ind» – инидикатор, показывающий, нарушается ли у хотя бы одной переменной требование пропорциональности, «glob» – аналогичный индикатор, показывающий, нарушается ли в целом для модели требование пропорциональности (нарушение требования пропорциональности может быть позже компенсировано введением зависящих от времени предикторов).

Таблица – Пример результата работы программного решения для перебора моделей выживаемости

formula	n	aic	aic0	R2_MF	R2_Nag	conc	χ2	df	p.value	ind	glob
Surv(time = v6, event = v5) ~ v10 + v11 + v19 + v22 + v24 + v29	188	146.5	161.6	0.127	0.668	0.845	207.0	11.000	0.000	1	1
Surv(time = v6, event = v5) ~ v10 + v11 + v12 + v19 + v22 + v29	198	145.7	163.9	0.117	0.621	0.796	192.2	11.000	0.000	0	0
Surv(time = v6, event = v5) ~ v8 + v10 + v11 + v19 + v22 + v29	198	146.6	163.9	0.115	0.614	0.798	188.3	11.000	0.000	1	1
Surv(time = v6, event = v5) ~ v10 + v11 + v19 + v24 + v27 + v29	188	147.5	163.6	0.108	0.610	0.822	177.0	11.000	0.000	0	1
Surv(time = v6, event = v5) ~ v10 + v11 + v19 + v22 + v29	198	147.2	163.9	0.113	0.607	0.798	184.7	10.000	0.000	1	1

Разработанное решение предполагается свободно распространять как пакет расширения языка «R». Ускорение расчётов возможно, как уже было указано как динамическую библиотеку и как пакет расширения. Программа будет полезна исследователям в области медицинской статистики при анализе данных пилотных исследований.

ЛИТЕРАТУРА

1. *Pölsterl, S.* Survival Analysis for Deep Learning [Electronic resource] / Sebastian Pölsterl. – Mode of access: https://k-d-w.org/blog/2019/07/survival-analysis-for-deep-learning/. – Date of access: 03.03.2020.

2. *Ogundimu, E. O.* Adequate sample size for developing prediction models is not simply related to events per variable / E.O. Ogundimu, D.G. Altman, G.S. Collins // J Clin Epidemiol. – 2016. – Vol. 76. – P. 175-182.

3. Armadillo: C++ library for linear algebra & scientific computing [Электронный ресурс]. – Режим доступа: http://arma.sourceforge.net/. – Дата доступа: 04.03.2020.

4. R Core Team. R: A Language and Environment for Statistical Computing [Электронный ресурс] : R. – Режим доступа: https://www.r-project.org/about.html. – Дата доступа: 01.05.2018.

5. *Eddelbuettel, D.* RcppArmadillo: «Rcpp» Integration for the «Armadillo» Templated Linear Algebra Library / D. Eddelbuettel, R. Francois, D.B. and B.N.R. author details. – 2020.