Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных систем
Информационно-аналитические системы

Марюфич Михаил Романович

Извлечение признаков из данных высокой размерности

Бакалаврская работа

Научный руководитель:
к. ф.-м. н., доцент Графеева Н.Г.

Рецензент:
Путин Е.О.

Санкт-Петербург
2017

SAINT-PETERSBURG STATE UNIVERSITY

Software and Administration of Information Systems
Analytical Information Systems

Mikhail Maryufich

Feature Selection for hight-dimensional data

Bachelor's Thesis

Scientific supervisor:
associate professor Natalia Grafeeva

Reviewer:
Eugene Putin

Saint-Petersburg
2017

Оглавление

Введение

В настоящее время человеческая цивилизация накопила огромный объем данных в самых различных сферах и это приводит к тому, что можно эффективно применять машинное обучение для множества задач. В некоторых задачах объекты имеют очень высокую размерность, что негативно сказывается на времени обучения. Так же, многие из признаков, описывающие объект на самом деле не являются сколько бы то ни было важными, на самом деле можно их исключить и это с высокой вероятностью положительно скажется на результате.

Итак, первый плюс извлечения признаков – ускорение сходимости алгоритмов машинного обучения, что делает их применимыми на практике.

Многие из методов не просто позволяют отобрать признаки по какому-то порогу, а отранжировать их. Определить какой вклад вносит тот или иной признак для решения задачи. Это позволяют лучше понимать понимать предметную область.

Пример: задача предсказания пола(возраста), наличия какого-либо заболевания по генной экспрессии. В подобных задачах очень много признаков(порядка 15000), в подобных задачах можно отобрать на порядок меньше признаков, на которых модели машинного обучения будут давать такое же значения по метрикам или выше, чем на оригинальном наборе и обучаться быстрее. Ученые, хорошо знающие предметную область могут посмотреть на топ признаков и понять на какие именно гены нужно изучить с биологической точки зрения).

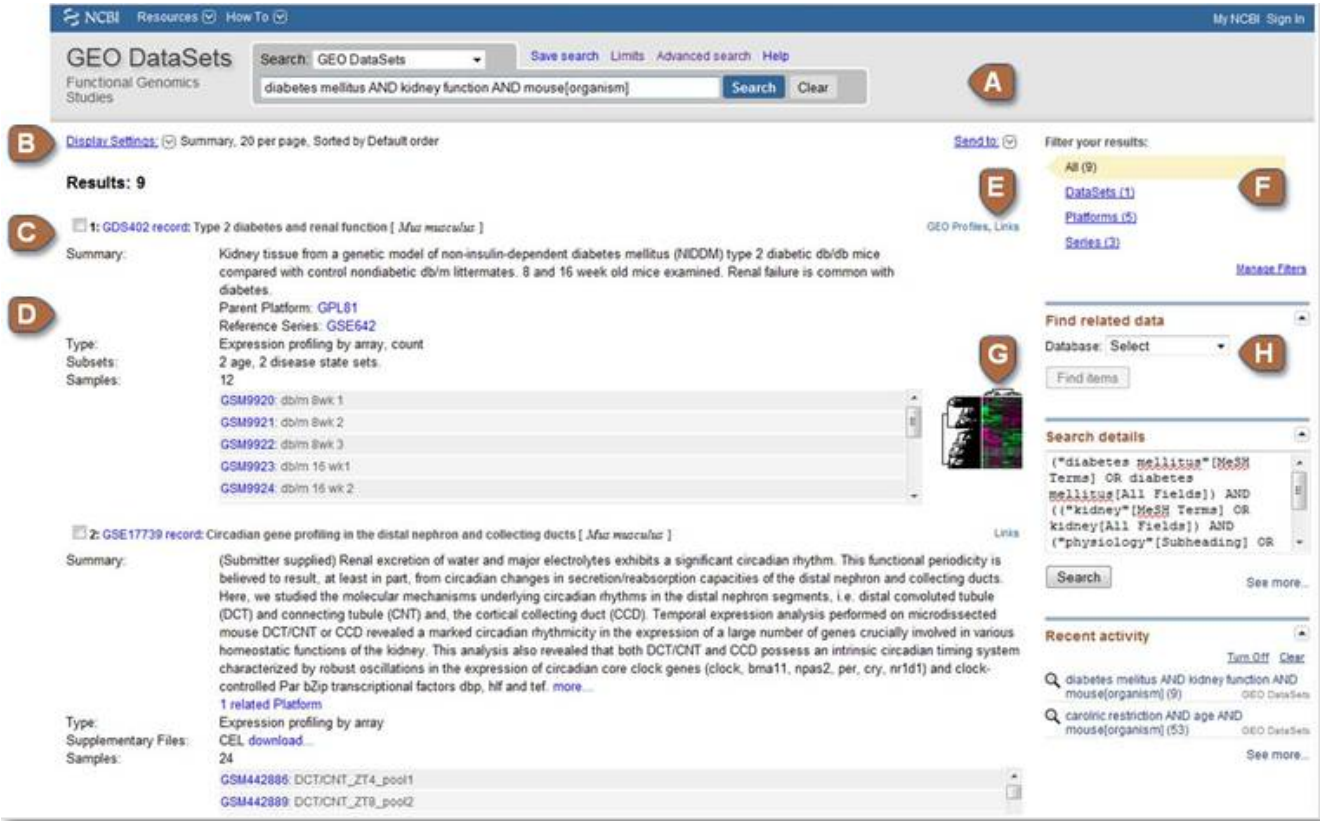
1. Описание данные

1.1. Реальные данные

Данные были взяты с [11]. Они представляют с собой таблицу с генной экспрессией по платформе GEO для предсказания пола.

Размерность признакового описания объектов – 12000.

Количество объектов – 40000(выборка сбалансирована)



2. Постановка задачи

2.1. Формальная постановка задачи

F – множество признаков.

Цель: выбрать наименьшее подмножество F, т.к что значение основных метрик для решаемой задачи машинного обучения будет не ниже чем на полном наборе.

2.2. Методика постановки эксперимента

Мы будем действовать в три этапа

1. Строим модель(модели) машинного обучения на полном пространстве признаков и оцениваем метрики.

2. Строим ранжирующий список, с помощью одного(или нескольних) из методов

3. Строим модели машинного обучения на извлеченном подпространстве и оцениваем метрики.

Рис. 1: Схема постановки эксперимента

2.3. Используемые метрики

Для задач классификации используют:

- Accuracy – можно использовать в случае целиком сбалансированной выборки
- F1-score – используется в большинстве работ, подходит для задач многоклассовой классификации и несбалансированных выборок.
- ROC-AUC – подходит для задач бинарной классификации и дает для них наиболее интерпретируемый результат(например для константного предсказания roc-auc будет = 0.5)

3. Описание методов извлечения признаков

3.1. Переборные методы

Первое, что приходит в голову – это просто перебрать все возможные наборы признаков. К сожалению, на данном уровне развития вычислительной техники это вычислительно невозможно как мощность перебираемого множества составит 2^n, где n – количество признаков.

Второе – посчитать метрики на каждом признаке отдельно, а затем жадно их сливать. Мощность перебираемого множества составит $n \log n$.) На практике такой алгоритм применяется довольно часто, но он не позволяет судить о априорной важности каждого признака и тоже вычислительно сложен.

3.2. Генетический алгоритм для извлечения признаков

1. Инициируем популяции случайным набором признаков и обучаем на каждой из них модель машинного обучения
2. Скрещиваем и мутируем популяции.
3. Оставляем top N популяции.
4. Повторяем до тех пор, пока не будет получен удовлетворяющий нас результат.

Опять таки данный подход не позволяет создать ранжированный список признаков и сходимость может занимать много времени.

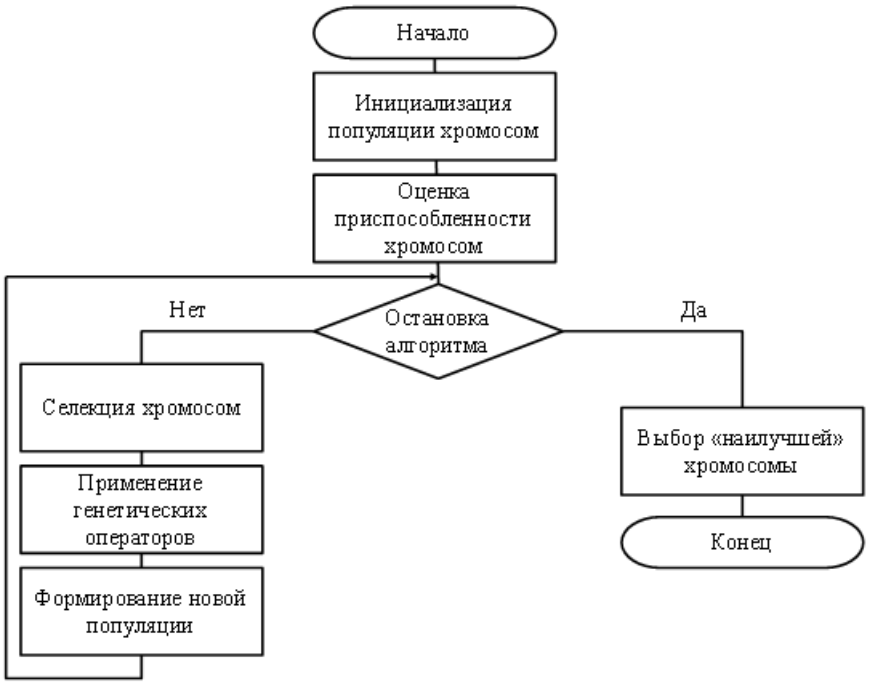


Рис. 2: Общая схема генетичекого алгоритма

3.3. Основанные на похожести признаков.

- ReliefF [7] – алгоритм выбора признаков, используемый в бинарной классификации (обобщаемый для полиномиальной классификации путем декомпозиции на ряд двоичных задач), предложенный Кирой и Ренделлом в 1992 году Его сильные стороны заключаются в том, что он не зависит от эвристики, работает в полиномиальном времени низкого порядка и является устойчивым к помехам и устойчивым к взаимодействию признаков, а также применим для двоичных или непрерывных данных; Однако, он не делает различий между избыточными функциями, а небольшое количество учебных экземпляров обманывает алгоритм.

- Fisher-Score [6] – является одним из наиболее широко используемых методов отбора признанных специалистов. Однако он выбирает каждую функцию независимо в соответствии с их оценками в соответствии с критерием Фишера, что приводит к субоптимальному подмножеству признаков

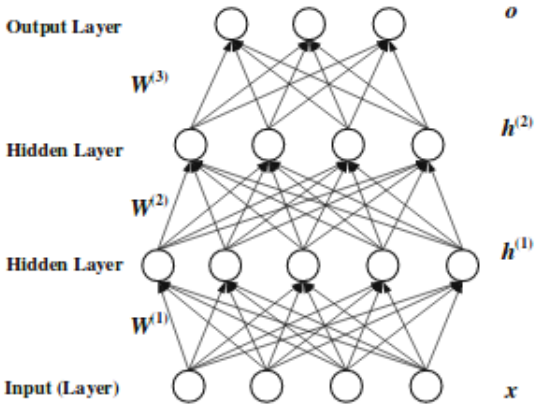
3.4. Как извлекать ранжирующие списки из моделей машинного обучения

- Random Forest – сортируем признаки по относительной частоте встречаемости в узловых условиях. [4]
- Логистическая регрессия – смотрим на коэффициенты в построенной модели. [12]
- Градиентный бустинг на решающих деревьях – аналогично Random Forest. [5]

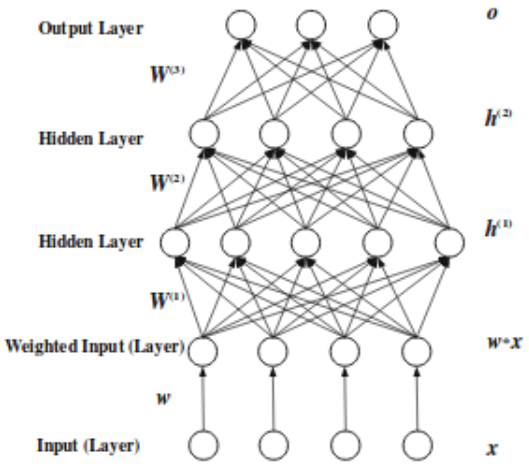
3.5. DFS – deep feature selection

Метод [8] предлагает новую архитектуру нейронных сетей, основанную и модифицирующую многослойный перцептрон, введением дополнительного слоя между входами и первым полносвязным слоем. Для построения ранжирующего списка нужно

- Обучить нейронную сеть с помощью алгоритма обратного распространения ошибки [3]
- Отсортировать веса нового слоя и использовать их как список относительной важности признаков.



(c) MLP



(d) Deep DFS

3.6. HVS

Впервые описан в [11].

Заключается в том, что мы считаем относительную важность i-го признака, нормируя веса следующим образом:

$$S_i = \sum_{j \in H} \left(\frac{|w_{ji}|}{\sum_{i' \in I} |w_{ji'}|} \sum_{k \in O} \frac{|w_{kj}|}{\sum_{j' \in H} |w_{kj'}|} \right)$$

3.7. Методы с производными

Описаны в данной статье [10].

Является наследником и расширением предыдущего метода метода.

x^l – i-й вход сети.

x_i – j-й выход(ответ) – следует обратить внимание, что выход может быть из R^n, то есть методы годятся в случае мультиклассовой классификации и тд.

N – количество сэмплов.

M – количество выходов нейронной сети.

S_i – относительная важности i-го признака.

Данные методы также эвритические и эвристика заключается в том, что хорошо учитывать производные сети по данным, где меняется сильнее – те важнее.

Ruck et al.	$S_i = \sum_{l=1}^{N} \sum_{j=1}^{M} \left\| \frac{df_j}{dx_i}(x^l) \right\|$
Refenes et al.	$S_i = \frac{1}{N} \frac{var(x_i)}{var(f(x) - y)} \sum_{l=1}^{N} \left(\frac{df}{dx_i}(x^l) \right)^2$
Refenes et al.	$S_i = \frac{1}{N^{\frac{1}{2}}} \frac{\left(\sum_{l=1}^{N} \left(\frac{df}{dx_i}(x^l) - \sum_{j=1}^{N} \frac{df}{dx_i}(x^j) \right)^2 \right)^{\frac{1}{2}}}{\sum_{l=1}^{N} \frac{df}{dx_i}(x^l)}$
Refenes et al.	$S_i = \frac{1}{N} \sum_{l=1}^{N} \left\| \frac{df}{dx_i}(x^l) \frac{x_i}{f(x^l)} \right\|$
Dorizzi et al.	$S_i = q_{95} \left(\left\| \frac{df}{dx_i}(x) \right\| \right)$
Czernichow et al.	$S_i = \frac{\sum_{l=1}^{N} \left(\frac{df}{dx_i}(x^l) \right)^2}{max_j \left(\sum_{l=1}^{N} \left(\frac{df}{dx_j}(x^l) \right)^2 \right)}$

3.8. Комбинация DFS и производных по входам

Данные методы даже по отдельности очень хорошо показывают себя на практике(если нейронную сеть возможно обучить).

Так как DFS – ни что иное, как просто новая архитектура, обучаемая обратным растранением ошибки, мы можем комбинировать вышеизложенные методы. Например, обучить DFS, потом получить ранжирующий список с помощью одного из методов с производными. В разделе эксперименты будет описано, как данный подход улучшает результат.

4. Эксперименты и результаты

Все эксперименты проводились на языке программирования python с использованием пакетов sklearn[9], skfeature, xgboost[1], keras [2].

Для реализации метода DFS – был написан свой слой для библиотеки keras.

HVS и методы, испольующие производные по входам так же были написаны с использованием этой библиотеки.

4.1. Построение базовых моделей для задачи предсказания пола

Оцениваем по метрике roc auc, так как данная метрика хорошо интерпретируема и позволяет понять легко понять, что одна модель лучше другой. Кроме того для сбалансированных выборок эти метрики очень сильно кореллируют между собой, поэтому по большему счету нет большой разницы какую из них использовать.

Для того, чтобы бороться с эффектом переобучения, часто возникающим на практике – используем 5-тифолдовуя кроссвалидацию.

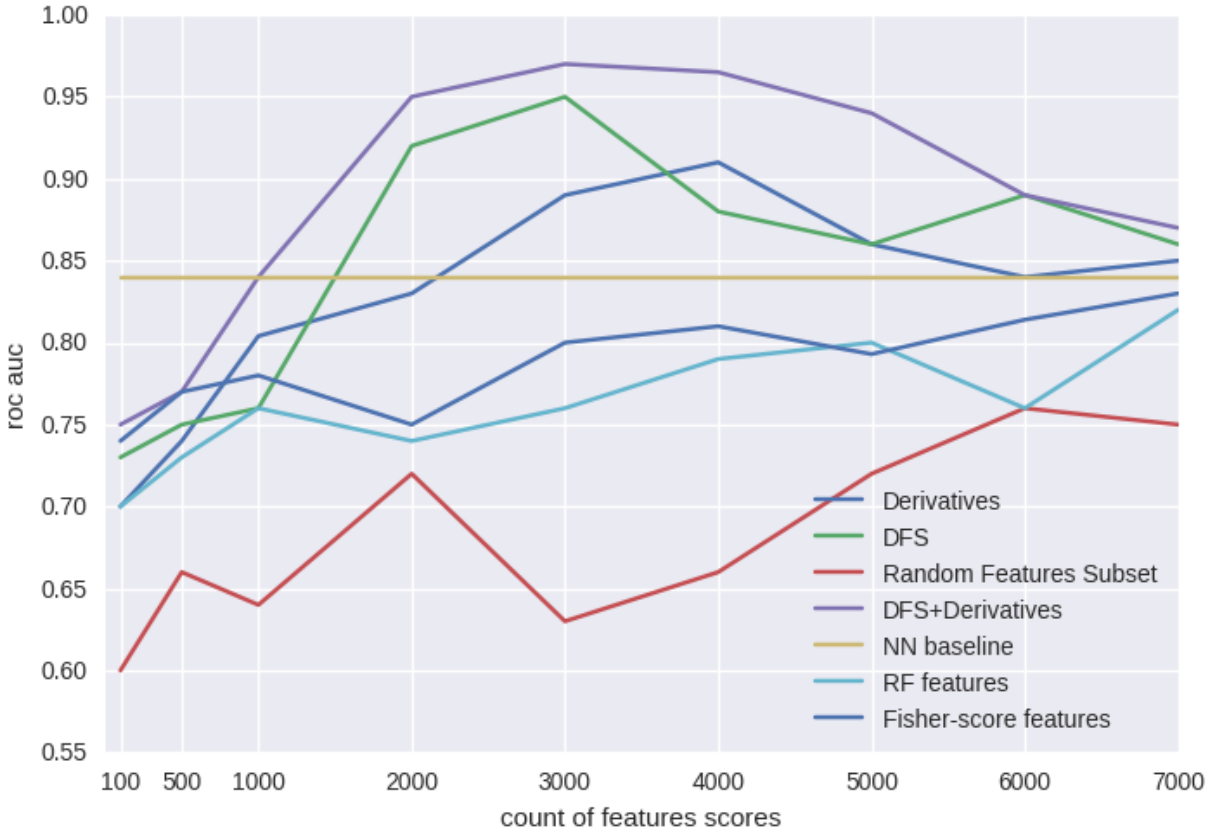
Получаем следующий результат

- Random Forest – 0.67
- Logistic Regression – 0.59
- Gradient Boosting Machine – 0.72
- Двухслойная нейросеть – 0.84
- Трехслойная нейросеть – 0.82

Как видно, на наших данные нейронные сети оказались лучше других моделей и это достаточно ожидаемо, так как на таких данных, где все признаки имеют одинаковый числовой характер модели, основанные на деревьях всегда показывают не лучший результат, а логистическая регрессия по построению не улавливает слишком сложных зависимостей.

4.2. Сравнительный анализ различных методов

Проводим эксперименты согласно методике, описанной в соответствующем разделе.

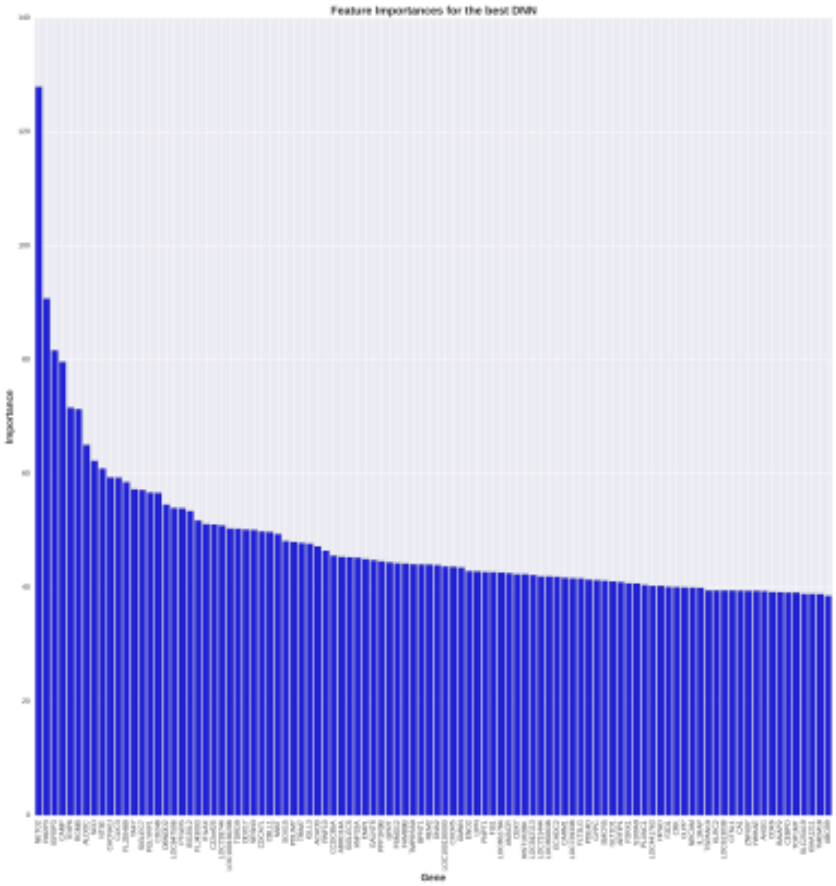


Видим, что придуманный метод, комбинирующий Deep Feature Selection и методы, основанные на производных получает результат, превосходящий остальные.

Лучший ROC AUC – 0.96.

Также удалось добиться результата, который сильно превосходит лучшую модель, построенную на полном пространстве, сжав пространство признаков в 6 раз.

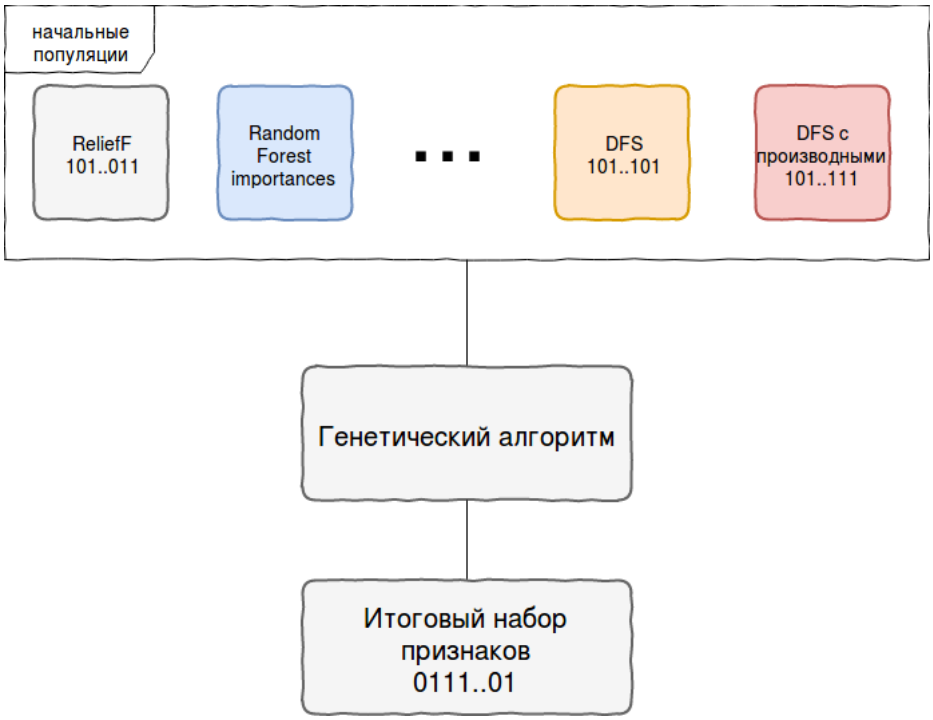
Кроме того. если проанализировать ранжированный список признаков:



То видно, что выделилось n признаков, которые явно имеют большой вес для получения результат. В будущем возможна работа по оценке их биологической значимости.

4.3. Улучшение с помощью генетического алгоритма

В результате экспементов мы сохранили списки важных генов для каждого из метода. Оказалось, что можно улучить значение метрики, использовав генетический алгоритм(описанный в соответствующем разделе), инициировав популяции самыми 1000-чью самыми важными их признаками и проведя 10 итераций алгоритма. В результате получили roc auc 0.97, улучшив предыдущий лучший результат на 0.01.



5. Заключение

В работе были описаны различные методы Feature Selection, каждый из них был применен на реальных данных. Был придуман и протестирован новый метод извлечения признаков, полученный комбинираванием DFS и методов, основанных на производных. Показана модификация генетического алгоритма для извлечения признаков.

6. Дополнительные материалы

6.1. Гиперпараметры алгоритмов машинного обучения

Оптимальные параметры подбирались с помощью метода перебора по сетке и оценивания метрики roc-auc на 5-ти фолдовой кроссвалидации.

- Random Forest
  - $n\_estimators$: 500
  - $criterion$: entropy
  - $min\_samples\_leaf$: 2
  - $min\_samples\_split$: 3
- Logistic Regression
  - $penalty$: l2
  - C: 0.01
- XGBoost
  - eta: 0.2
  - $alpha$: 0.1
  - $lambda$: 0.9
- Двухслойная нейронная сеть
  - $optimizer$: Adam, lr: 0.001
  - $dropout$: 0.2
  - $Activation$: Relu
  - На первом скрытом слое – 1000 нейронов, на втором – 500
- Треслойная нейронная сеть

– *optimizer*: Adadelta, lr: 0.001

– *dropout*: 0.2

– *Activation*: Relu

– На первом скрытом слое – 1000 нейронов, на втором – 300, на третьем – 500(используется техника бутылочного горлышка)

Список литературы

[1] Chen Tianqi, Guestrin Carlos. XGBoost: A Scalable Tree Boosting System // CoRR. — 2016. — Vol. abs/1603.02754. — URL: http://arxiv.org/abs/1603.02754.

[2] Chollet François et al. Keras. — https://github.com/fchollet/keras. — 2015.

[3] Effiicient BackProp / Yann LeCun, Léon Bottou, Genevieve B. Orr, Klaus-Robert Müller // Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop. — London, UK, UK : Springer-Verlag, 1998. — P. 9–50. — URL: http://dl.acm.org/citation.cfm?id=645754.668382.

[4] Genuer Robin, Poggi Jean-Michel, Tuleau-Malot Christine. Variable Selection Using Random Forests // Pattern Recogn. Lett. — 2010. — . — Vol. 31, no. 14. — P. 2225–2236. — URL: http://dx.doi.org/10.1016/j.patrec.2010.03.014.

[5] Gradient Boosted Feature Selection / Zhixiang Xu, Gao Huang, Kilian Q. Weinberger, Alice X. Zheng // Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. — KDD '14. — New York, NY, USA : ACM, 2014. — P. 522–531. — URL: http://doi.acm.org/10.1145/2623330.2623635.

[6] Gu Quanquan, Li Zhenhui, Han Jiawei. Generalized Fisher Score for Feature Selection // CoRR. — 2012. — Vol. abs/1202.3725. — URL: http://arxiv.org/abs/1202.3725.

[7] Kira Kenji, Rendell Larry A. The Feature Selection Problem: Traditional Methods and a New Algorithm // Proceedings of the Tenth National Conference on Artificial Intelligence. — AAAI'92. — AAAI Press, 1992. — P. 129–134. — URL: http://dl.acm.org/citation.cfm?id=1867135.1867155.

[8] Li Yifeng, Chen Chih-Yu, Wasserman Wyeth W. Deep Feature Selection: Theory and Application to Identify Enhancers and Promoters // Research in Computational Molecular Biology: 19th Annual International Conference, RECOMB 2015, Warsaw, Poland, April 12-15, 2015, Proceedings / Ed. by Teresa M. Przytycka. — Cham : Springer International Publishing, 2015. — P. 205–217. — ISBN: 978-3-319-16706-0. — URL: http://dx.doi.org/10.1007/978-3-319-16706-0_20.

[9] Scikit-learn: Machine Learning in Python / F. Pedregosa, G. Varoquaux, A. Gramfort et al. // Journal of Machine Learning Research. — 2011. — Vol. 12. — P. 2825–2830.

[10] Verikas A., Bacauskiene M. Feature Selection with Neural Networks // Pattern Recogn. Lett. — 2002. — . — Vol. 23, no. 11. — P. 1323–1335. — URL: http://dx.doi.org/10.1016/S0167-8655(02)00081-8.

[11] Yacoub M., Bennani Y. HVS: A Heuristic for Variable Selection in Multilayer Artificial Neural Network Classifier. — Intelligent Engineering Systems through Artificial Neural Networks, St. Louis, 1997.

[12] Zhang Zhongheng. Variable selection with stepwise and best subset approaches // Annals of Translational Medicine. — 2016. — Vol. 4, no. 7. — URL: http://atm.amegroups.com/article/view/9706.