Домашнее задание 5

Авласов Владислав

Задание 1

В алгоритме k-fold на что влияет параметр k? В каких случаях стоит выбирать его большим, в каких маленьким?

k влияет на несколько аспектов:
1. На время работы нашего алгоритма. Нам нужно запустить алгоритм k раз, чтобы использовать каждую из k подгрупп в качестве тестовой выборки. Если данных много, то брать очень большое k не выгодно из-за временных затрат.
2. Объективность оценки тестовой выборки. Если взять k большим, тестовая выборка каждый раз будет очень маленькой. При k близком к N(количество объектов) она может даже не превосходить VC-dim нашей гипотезы, т.е. результаты на ней вообще ничего не скажут. Из-за того, что мы не знаем распределения D, всё же желательно выбирать k так, чтобы статистически тестовая выборка была репрезентативна. Также при большом k выборки отличаются слабо одна от другой, поэтому полученные оценки имеют высокий коэффициент корреляции. Для некоторых задач есть исключения в виде leave-one-out, но это частный случай.
3. Качество обучения. Если взять k маленьким, например $k = 2$, то наша модель будет недоучиваться, т.к. половину выборки мы у неё заберём. Особенно не выгодно в случае, если размер самой выборки не велик.

Выбор k – дискуссионый вопрос, надо отталкиваться от технических ресурсов, выборки, задачи и т.д.

Задание 2

Почему в практических задачах делят выборку на три части: тренировочную, валидационную и тестовую? Почему не хватает первых двух?

На валидационной выборке мы проверяем гипотезы и выбираем минимальную оценку, по сути подгоняем алгоритм под эту выборку. И не очень хорошо, отталкиваясь от этой оценки, выбирать окончательный алгоритм. Поэтому люди оставляют(откладывают) ещё какую-то часть данных на последнюю проверку, так сказать, *перед продакшном*, когда уже выбрали гипотезу и её параметры, чтобы получить независимую и более честную оценку. На тестовой выборке получают true-risk, и на его основе уже принимают решение, оставить гипотезу или же всё начать сначала. Чтобы не было переобучения, которое возможно на валидационной выборке, тестовую выборку нельзя использовать много раз. Если было решено искать новую гипотезу после запуска на тестовой выборке, то для следующей гипотезы нужно добыть новую тестовую выборку, каким-либо образом достав новые данные.

Задание 3

Рассмотрим альтернативу алгоритму k-fold алгоритм "leave-all-out". В этом алгоритме мы раз- биваем всеми возможными способами выборку на две части на одной части тренируем алгоритм, а на другой (отложенной) измеряем качество. Затем все полученные измерения усредняем и используем в качестве оценки L(D). В чём преимущества и недостатки данного подхода по сравнению с k-fold?

Это очень трудоёмкий алгоритм, т.к. нам придётся запуститься $\sum_{k=1}^{n} C_n^k$ раз, т.е. это экспоненциально много раз по отношению к размеру выборки. Даже на сравнительно небольших данных алгоритм будет работать вечно. Не могу найти преимущества данного подхода. Возможно, с теоретической точки зрения такая стратегия позволит всесторонне изучить поведение нашего алгоритма и гипотез, тем самым дав нам исчерпывающую оценку эффективности алгоритма, но на текущем техническом уровне не применимо для практических задач.