Сравнение разных подходов в решении Multi-armed bandit problem.

ЗАХАРОВ ГЕОРГИЙ, Московский физико-технический институт, Россия

Поиск наилучшего эксперта является классической задачей онлайн оптимизации для поиска наилучшей модели. Наша цель описать подходы к решению этой задачи, найти для них оптимальные параметры и сравнить между собой.

1 ОПИСАНИЕ АЛГОРИТМОВ

Будем считать, что количество экспертов n, а количество итераций k.

Кратко опишем алгоритмы.

Binary weights majority vote: Первая реализация будет выбирать вариант за который проголосовала хотя бы половина экспертов. Если предсказание окажется неверным, то всех кого мы послушали – выгоняем. Повторяем процесс пока у нас не останется только один эксперт. Тогда уже делать нечего – слушаем только его. Проще говоря мы выбираем голос большинства с весами, но веса у нас бинарные: либо 1, либо 0.

Основная идея в том, что при ошибке мы избавляемся от хотя бы половины экспертов, а значит алгоритм сойдется к какому-то решению за не более чем $log_2(n)$ к некоторому решению.

Decreasing weights majority vote: Вторая реализация тоже будет выбирать вариант за который проголосовала большинство экспертов, только в этот раз мы будем поддерживать ненулевые веса у всех экспертов. А именно мы будем уменьшать вес тех, кто ответил неверно после каждой итерации, так как мы хотим реже слушать тех, кто чаще ошибается. Веса будут обновляться по формуле:

$$w_{i+1} = w_i(1 - I_{[p_i=0]}\epsilon)$$

где p_i – ответ i–го эксперта, а ϵ – гиперпараметр. Стандартный параметр = 0.1, а оптимальный будет в точности равен $min(\sqrt{\frac{(n+1)\cdot ln(n+1)}{k}}, 0.5)$.

Для этого алгоритма выполняется оценка: Пусть m_i^t — количество ошибок допущенных i–ым экспертом за t шагов, а m^t — количество ошибок допущенных нашим алгоритмом за t шагов. Тогда для $\epsilon < 0.5$ верно:

$$m^t \leq \frac{2lnn}{\epsilon} + 2(1+\epsilon)m_i^t, \forall i$$

Докажем это. Заметим, что на t-ой итерации $w_i^t = (1-\epsilon)^{m_i^t}$. Пусть $A^t = \Sigma w_i^t$. Тогда $A^1 = n$. Каждый раз, когда алгоритм ошибается мы умножаем на $1-\epsilon$ хотя бы половину весов. Тогда A^t оценивается рекурсивно в случае ошибки:

$$A^t \leq A^{t-1}(\frac{1}{2} + \frac{1-\epsilon}{2}) =\leq A^{t-1}(1 - \frac{\epsilon}{2}) \leq A^1(1-\frac{\epsilon}{2})^{m^t} = n(1-\frac{\epsilon}{2})^{m^t}$$

В то же время мы хотим достигнуть точности лучшего из экспертов, потому оценим $w_i^t \leq A^t$ одним слагаемым. Как следствие

$$(1-\epsilon)^{m_i^t} = w_i^t \leq n(1-\frac{\epsilon}{2})^{m^t}$$

Прологарифмируем это неравенство

$$m_i^t \cdot ln(1-\epsilon) \leq ln(n) + m^t \cdot ln(1-\frac{\epsilon}{2})$$

Author's address: Захаров Георгий, zakharov.ge@phystech.edu, Московский физико-технический институт, Москва, Россия.

Оценим $x \leq -ln(1-x) \leq x + x^2$, при $0 \leq x \leq 0.5$ тогда

$$m^t \cdot \epsilon \leq -m^t \cdot ln(1 - \frac{\epsilon}{2}) \leq ln(n) + m_i^t(\epsilon + \epsilon^2)$$

А значит

$$m^t \leq \frac{ln(n)}{\epsilon} + m_i^t(1+\epsilon)$$

Остается заметить, что оптимальным параметром является такой, что $n-$ая m_i^t — распределение ошибки $i-$ой порядковой статистики. Статистику можно оценить ее матожиданием. Для равномерного это будет в точности $\frac{n}{n+1}$. Тогда минимумом кавдратного трехчлена будет в точности $\sqrt{\frac{(n+1)\cdot ln(n+1)}{k}}$.

Randomized weights majority vote: Третья реализация также требует пересчет весов, только при ошибке мы будем уменьшать вес, а при успехе наоборот − увеличивать. Будем считать, что $M(i,t)$ — функция выигрыша $i-$го эксперта на исходе $t-$ой операции, а $M(D,t)$ — наша функция выигрыша. Для простоты будем считать, что если эксперт оказался прав, то его выигрыш в точности равен 1, а если не прав, то -1. Веса будут обновляться по формуле:

$$w_{t+1} = \begin{cases} w_t(1-\epsilon)^{M(i,t)} & \text{, если эксперт ошибся} \\ w_i(1+\epsilon)^{-M(i,t)} & \text{, если эксперт оказался прав} \end{cases}$$

Заметим, что если бы мы, например, считали, что ошибка эксперта дает выигрыш 0, а не -1, то формула пересчета совпадала с предыдущим пунктом. Теперь вместо того чтобы брать сумму взвешенных весов, попробуем выбирать эксперта случайно основываясь на его весе, то есть теперь алгоритм будет рандомизированный.

Будем считать, что $M(i,t)$ — функция выигрыша $i-$го эксперта на исходе $t-$ой операции, а $M(D,t)$ — наша функция выигрыша. Тогда для $\epsilon < 0.5$ верно:

$$M(D,t) \leq \frac{ln(n)}{\epsilon} + (1+\epsilon)\Sigma_t M(i,t) I_{[M(i,t)0]} + (1-\epsilon)\Sigma_t M(i,t) I_{[M(i,t)<0]}, \forall i$$

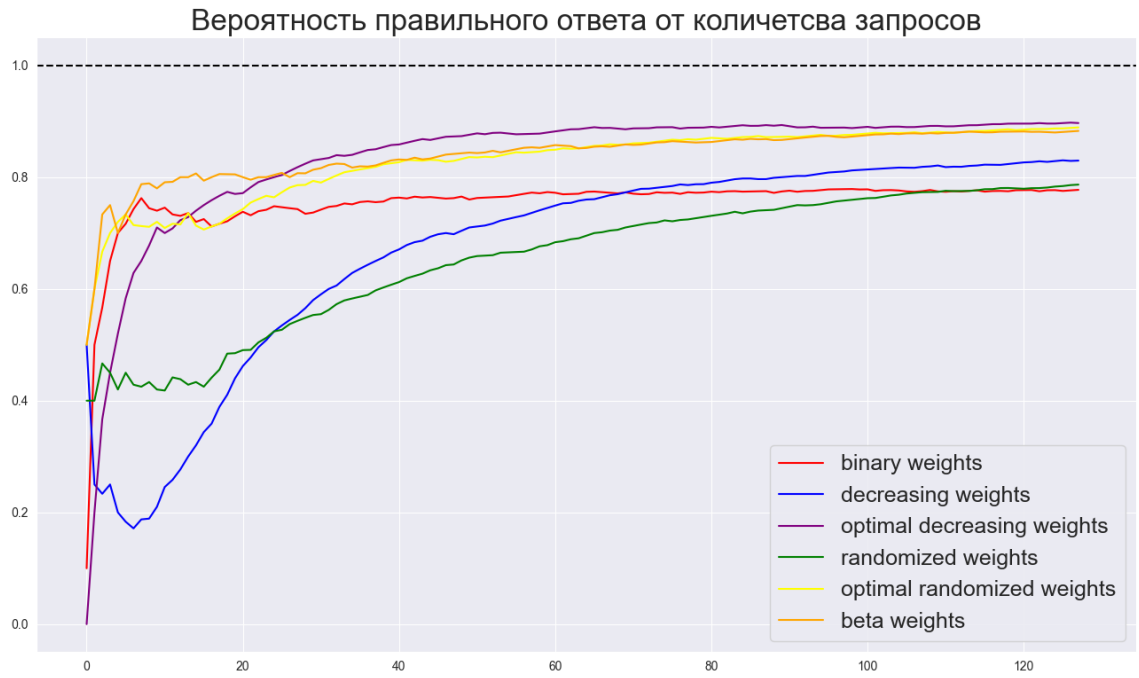
Доказывается это утверждение аналогичной техникой что и в предыдущем алгоритме, используя неравенство Бернулли $(1-\epsilon)^x \leq (1-\epsilon x)$, $(1+\epsilon)^{-x} \leq (1-\epsilon x)$.

Beta weights majority vote: Последний алгоритм будет делать предположения основываясь на апостериорном для каждого эксперта распределении Бернулли. Мы будем пытаться предугадать распределение каждого отдельного эксперта. При этом параметры $Beta-$распределения обновляются так: если эксперт ошибся, то $\alpha \to \alpha + 1$, а $\beta \to \beta$, а если эксперт оказался прав, то $\alpha \to \alpha$, а $\beta \to \beta + 1$. Теперь, чтобы сделать предсказание, мы будем брать среднее арифметическое от апостериорных распределений каждого эксперта. Если среднее арифметическое больше 0.5, то мы будем предсказывать 1, иначе 0.

Основная идея в том, что апостериорное распределение способно приблизить оценку достаточно быстро, что позволяет находить предсказание быстрее.

2 РЕЗУЛЬТАТЫ

Подход с бинарными весами оказался одним из наиболее плохих. Его проблема ясна сразу: он отсекает экспертов на слишком раннем этапе, потому чаще всего сходится не к оптимальному среди них. Неоптимизированые версии *randomized weights* и *decreasing weights* показали себя довольно слабо, в то же время как их оптимизированные версии, а также *beta weights* достигают более высоких результатов, притом именно *optimal decreasing weights* сходится раньше всех.


Вероятность правильного ответа от количетсва запросов
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80
100
120
binary weights
decreasing weights
optimal decreasing weights
randomized weights
optimal randomized weights
beta weights