Задание №1.5. Классификация в условиях неполноты информации

В этом задании мы попробуем обучить бинарный классификатор в ситуации, когда нам известны метки (классы) не для всех примеров из обучающей выборки. В частности, мы рассмотрим случай, который не является таким уж редким на практике, при котором нам даны метки только некоторого подмножества положительных примеров. Все отрицательные примеры и оставшаяся часть положительных примеров меток не имеют.

Формализуем этот сценарий следующим образом. Пусть $\left\{\left(x^{(i)}, t^{(i)}\right)\right\}_{i=1}^{n}$ – это обычная выборка независимых одинаково распределенных примеров. Здесь $x^{(i)}$ – это входные данные (вектора признаков), а $t^{(i)}$ – метки. Теперь рассмотрим ситуацию, когда $t^{(i)}$ нам неизвестны, а вместо этого нам доступна информация о метках только для части положительных примеров. Более конкретно – у нас есть значения $y^{(i)}$, которые сгенерированы следующим распределением:

$$\begin{aligned}
&\forall x, p\left(y^{(i)} = 1 \mid t^{(i)} = 1, x^{(i)} = x\right) = \alpha,\\
&\forall x, p\left(y^{(i)} = 0 \mid t^{(i)} = 1, x^{(i)} = x\right) = 1 - \alpha,\\
&\forall x, p\left(y^{(i)} = 1 \mid t^{(i)} = 0, x^{(i)} = x\right) = 0,\\
&\forall x, p\left(y^{(i)} = 0 \mid t^{(i)} = 0, x^{(i)} = x\right) = 1,
\end{aligned}$$

где $\alpha \in (0,1)$ – неизвестный скаляр. Иными словами, если неизвестная нам «истинная» метка $t^{(i)}$ равна 1, то с вероятностью α мы имеем $y^{(i)} = 1$. С другой стороны, если неизвестная нам «истинная» метка $t^{(i)} = 0$, то мы всегда имеем $y^{(i)} = 0$.

Наша конечная цель – построить бинарный классификатор h истинной метки t в том случае, когда мы имеем доступ только к частичным меткам y. Другими словами, мы хотим, используя только x и y, построить h таким образом, что соотношение $h\left(x^{(i)}\right) \approx p\left(t^{(i)} = 1 \mid x^{(i)}\right)$ выполнялось как можно точнее.

Пример из реальной жизни: Предположим, мы ведем базу данных протеинов, которые участвуют в передаче сигналов через мембрану. Каждый образец, добавленный в базу данных, обладает этим свойством, но есть много таких, которые участвуют в межмембранной передаче сигналов, но еще отсутствуют в базе. Было бы полезно обучить классификатор идентифицировать протеины, которые можно рассматривать как кандидатов на добавление в базу данных. В наших обозначениях образец $x^{(i)}$ соответствует протеину, если $y^{(i)} = 1$, то протеин есть в базе данных, если $y^{(i)} = 0$, то он там отсутствует, и, наконец, $t^{(i)} = 1$ означает, что протеин участвует в межмембранной передаче сигналов, а $t^{(i)} = 0$ – что не участвует.

Для всех подзадач мы будем использовать следующие выборки и стартовый код:

- `train.csv, valid.csv, test.csv`
- `posonly.py`

Каждый файл содержит следующие столбцы: x_1, x_2, y и t. Каждая строка соответствует одному примеру. Значения $y^{(i)}$ сгенерированы процессом, описанным выше, с каким-то неизвестным α.

Вопрос №1. Задача на программирование: идеальный случай (все известно) [1 балл]

Сначала рассмотрим гипотетический (и не очень интересный) случай, когда у нас есть истинные значения меток для всей обучающей выборки (т.е. все значения $t^{(i)}$). В файле posonly.py допишите код, который обучит классификатор на основе логистической регрессии, используя x_1 и x_2 в качестве признаков и t в качестве меток. Пока игнорируйте значения y. Сохраните результаты работы модели на **тестовом** множестве в файл, указанный в коде.

Визуализируйте значения **тестовой** выборки на графике, в котором по горизонтальной оси идут значения x_1, а по вертикальной – x_2. Используйте разные символы, чтобы обозначать примеры разных классов. На этом же графике нарисуйте красным цветом решающую границу, найденную вашей моделью (т.е. прямую, соответствующую прогнозам с вероятностью 0.5).

Вопрос №2. Задача на программирование: наивный метод на подмножестве меток [1 балл]

Теперь рассмотрим случай, когда t-метки нам недоступны и мы можем проводить обучение, используя только значения y. Расширьте свой код в posonly.py, переобучив классификатор (по-прежнему используя x_1 и x_2 в качестве признаков), но только теперь используйте лишь y-метки. Сохраните результаты работы модели на **тестовом** множестве в соответствующий файл, указанный в коде.

Визуализируйте значения **тестовой** выборки на графике, в котором по горизонтальной оси идут значения x_1, а по вертикальной – x_2. Используйте разные символы, чтобы обозначать примеры $x^{(i)}$ с **истинной** меткой $t^{(i)} = 1$ (хотя при обучении мы использовали $y^{(i)}$), и чтобы обозначать примеры с $t^{(i)} = 0$. На этом же графике нарисуйте красным цветом решающую границу, найденную вашей моделью (т.е. прямую, соответствующую прогнозам с вероятностью 0.5).

Обратите внимание, что ваш алгоритм должен найти функцию гипотезы $h(\cdot)$, которая примерно предсказывает вероятность $p\left(y^{(i)} = 1 \mid x^{(i)}\right)$. Также заметим, что вполне ожидаемо эта гипотеза будет плохо предсказывать ту вероятность, которая нам на самом деле нужна, а именно $p\left(t^{(i)} = 1 \mid x^{(i)}\right)$, потому этот классификатор мы и называли наивным.

Далее мы попробуем улучшить наш наивный классификатор. Еще раз сформулируем задачу: имея доступ к выборке $\left\{\left(x^{(i)}, y^{(i)}\right)\right\}_{i=1}^{n}$, научиться хорошо предсказывать вероятность $p\left(t^{(i)} = 1 \mid x^{(i)}\right)$.

Вопрос №3. Правило Байеса [1 балл]

Покажите, что при сделанных нами предположениях, для любого i:

$$p\left(t^{(i)} = 1 \mid y^{(i)} = 1, x^{(i)}\right) = 1.$$

То есть, значение частичной метки $y^{(i)} = 1$ достоверно сообщает нам, что истинная метка тоже равна 1. Используйте правило Байеса, чтобы формально это доказать.

Вопрос №4 [1 балл]

Покажите, что при сделанных нами предположениях, вероятность того, что истинная метка $t^{(i)}$ является положительной, равна $1/\alpha$, умноженное на вероятность того, что частичная метка $y^{(i)}$ является положительной. То есть:

$$p\left(t^{(i)} = 1 \mid x^{(i)}\right) = \frac{1}{\alpha} \cdot p\left(y^{(i)} = 1 \mid x^{(i)}\right). \tag{1}$$

Заметим, что это означает, что если нам известно значение α,то мы можем преобразовать функцию, которая примерно предсказывает вероятность $h\left(x^{(i)}\right) \approx p\left(y^{(i)} = 1 \mid x^{(i)}\right)$ в функцию, которая примерно предсказывает вероятность $p\left(t^{(i)} = 1 \mid x^{(i)}\right)$, просто умножив ее на $1/\alpha$.

Вопрос №5. Оценка α [2 балла]

Стратегия для нахождения нужной нам вероятности $p\left(t^{(i)} \,\middle|\, x^{(i)}\right)$, обозначенная в предыдущем вопросе, требует нахождения α, которое нам неизвестно. Построим метод оценки значения α, основанный на использовании функции $h(\cdot) \approx p\left(y^{(i)} = 1 \,\middle|\, x^{(i)}\right)$, которую мы получили, решая вопрос №2.

Для того чтобы упростить анализ, предположим, что мы волшебным образом получили функцию $h(x)$, которая не примерно, а **точно** предсказывает эту вероятность: $h\left(x^{(i)}\right) = p\left(y^{(i)} = 1 \,\middle|\, x^{(i)}\right)$.

Далее сделаем принципиальное предположение, что $p\left(t^{(i)} = 1 \mid x^{(i)}\right) \in \{0,1\}$. Это означает, что в процессе, генерирующем истинные метки $t^{(i)}$, отсутствует шум. Это не такое уж оторванное от реальности предположение (в примере выше это означает, что либо белок обладает нужным нам свойством, либо нет; ситуацию, при которой одна молекула белка обладает свойством, а другая точно такая же вдруг его «теряет», мы не рассматриваем). При этом мы **НЕ** делаем предположения, что наблюдаемые метки $y^{(i)}$ тоже лишены шума – это было бы как раз необоснованно (то, что мы включили белок в базу данных, еще не означает, что он в действительности обладает нужным нам свойством – у нас может быть погрешность в детектировании этого свойства).

Теперь покажите, что

$$\alpha = \mathbb{E}\left[h\left(x^{(i)}\right) \,\middle|\, y^{(i)} = 1\right]. \qquad (2)$$

Полученный результат дает нам возможность оценить значение α, оценив правую часть указанного равенства. Пусть V_+ – это множество помеченных (и, соответственно, положительных) примеров валидационной выборки V, т.е. $V_+ = \left\{x^{(i)} \in V \,\middle|\, y^{(i)} = 1\right\}$. Тогда мы используем

$$\alpha \approx \frac{1}{|V_+|} \sum_{x^{(i)} \in V_+} h\left(x^{(i)}\right)$$

в качестве оценки α. Сам алгоритм вы реализуете в следующей подзадаче, а пока вам нужно только доказать корректность отношения (2).

Вопрос №6. Задача на программирование [1 балл]

Оцените константу α, взяв среднее от значений, возвращаемых вашим классификатором на всех положительных примерах валидационной выборки[1]:

$$\alpha \approx \frac{1}{|V_+|} \sum_{x^{(i)} \in V_+} h\left(x^{(i)}\right).$$

Добавьте код в `posonly.py`, масштабирующий прогнозы гипотезы $p\left(y^{(i)} = 1 \,\middle|\, x^{(i)}\right)$ вашего классификатора, обученного в подзадании 2, с помощью формулы (1) и полученного α.

Визуализируйте значения **тестовой** выборки на графике, в котором по горизонтальной оси идут значения x_1, а по вертикальной – x_2. Используйте разные символы, чтобы обозначать примеры $x^{(i)}$ с истинной меткой $t^{(i)} = 1$ и чтобы обозначить примеры с $t^{(i)} = 0$. На этом же графике нарисуйте красным цветом решающую границу, найденную вашей моделью (т.е. прямую, соответствующую **обновленным** прогнозам с вероятностью 0.5).

---

[1] Нет никакой очевидной причины, почему мы должны оценивать α, используя валидационную выборку вместо обучающей. Тем не менее, разница есть, и мы опускаем этот вопрос на самостоятельное изучение.

Замечание. Заметьте, что истинная вероятность $p(t \mid x)$ отличается от прогнозируемой $p(y \mid x)$ на константный множитель. Это означает, что если бы нашей целью было только проранжировать образцы в каком-нибудь определенном порядке (например, отсортировать в порядке убывания вероятности обладания заданным свойством), то в этом случае нам даже не нужно оценивать α, так как порядок, задаваемый $p(y \mid x)$, будет соответствовать порядку, задаваемому $p(t \mid x)$.