Семинар: вероятностная постановка задачи в машинном обучении. Логистическая регрессия

Федоров Артем Максимович

September 2024

1 Введение

Машинное обучение является одним из инструментов аналитиков, предназначенным для решения задач конкретной «сигнатуры». Интуитивно, каждый из нас, основываясь на личном опыте, способен сказать «данная проблема подлежит решению через ML» или обратное. Тем не менее, оставляя все на таком эмпирическом уровне мы обрекаем всю дальнейшую теорию на хаос. Поэтому данный семинар призван познакомить читающего с основами вероятностной парадигмы машинного обучения, формализовать постановку задачи для такого случая и дать интуицию, как и зачем это применяется.

Для начала логически выведем генезис задач машинного обучения:

1. Пусть задана система объектов Ω, изучение которых нам предстоит проводить в рамках решения. Такие объекты должны быть интерпретируемыми (смотря на объект $\omega \in \Omega$ мы должны быть способны измерить его, получить численные характеристики), но при этом не обязаны использоваться напрямую.

2. Для таких объектов из Ω определим множество функций $\mathrm{g}_i : \Omega \to \mathbb{R}$, что получают численные оценки на объекты, в последствии называемые features или же признаками. Важным замечанием здесь является то, что такие функции заданы на всем множестве Ω, что делает нашу модель порождения данных однородной.

3. В задаче требуется восстановить закон, по которому из наших наблюдаемых знаний об объекте ω (что в нашем случае является вектором наблюдений $x = \vec{\mathrm{g}}(\omega) = \{\mathrm{g}_1(\omega), \ldots, \mathrm{g}_k(\omega)\}$) определяется сокрытая информация. Фактически требуется построить стратегию $q : \mathrm{g}(\Omega) \to K$, где K есть множество целевых переменных.

4. В самой задаче определена выборка $(X_N, K_N) = \{(x_1, k_1) \ldots, (x_N, k_N)\}$. Требуется решить задачу $\mathcal{R}(q) \longrightarrow \min\limits_{q \in M}$ минизации риска по стратегиям из некоторого класса.

Пример 1

Задача: предсказать потребность в ремонте станка

- Ω – все возможные состояния станка
- g_i – данные снятые с датчика
- $K = \{0, 1\}$ – сломается ли станок в ближайшую неделю

Пример 2

Задача: определить ущерб после стихийного бедствия

- Ω – стихийное бедствие
- g_i – его параметры (место, время, оценка силы ...)
- $K = [0, \infty]$ – стоимость ущерба инфраструктуре и частной собственности

Пример 3

Задача: по EEG и MEG человека определить приказ для робота-манипулятора

- Ω – все возможные мысли человека
- g_i – получаемые временные ряды с каждого датчика
- $K = \mathit{Actionspace}$ – пространство действий (схватить, повернуть влево, сдвинуться)

2 Вероятностная парадигма машинного обучения

Пример: Рассмотрим задачу определения пола человека по его фотографии. Для этого будет подразумеваться, что была определена случайная выборка людей (например с улицы) и фотосъемка проводилась в разных условиях (погода, разные камеры, разная одежда на людях) - каждый рецензент проходил ровно один раз. Тогда $\Omega = \Omega_1 \times \Omega_2$ - все возможные обстоятельства, влияющие на пол человека и его внешний вид, повлиявшие на вид человека, а так же все условия, влияющие на фотографию человека. K – есть пол человека.

Очевидно, что мы можем построить прообраз каждого из классов в пространстве Ω и в пространстве наблюдений $\vec{g}(\Omega)$, как все такие $\omega : \vec{g}(\omega)$ – мужчины или женщины (рис. 1). Точно так же в получившейся модели порождения данных каждое наблюдение может быть порождено не единичным элементом $\omega \in \Omega$, так как не все наблюдения зависят от всех возможных событий в мире. Тогда получаем: $\exists\, \mathcal{U} = \mathrm{g}^{-1}(x)$ множество всех ω, что результируют в получении наблюдения x, при этом из эмпирического опыта нам известно, что на фото существуют мужчины, похожие на женщин, и женщины на мужчин. Оказывается, мы не можем построить детерминированный полностью правильный классификатор, так как множество $\mathcal{U}$ не обязательно вложено в M или W, мы лишь предполагаем, что для $K = M \Rightarrow \mathbb{P}(M|U) >> \mathbb{P}(W|U)$ и наоборот, где $\mathbb{P}$ мера множества. Таким образом мы стараемся недостаток в наблюдениях восполнить анализом прообразов, оценкой условных вероятностей множеств.

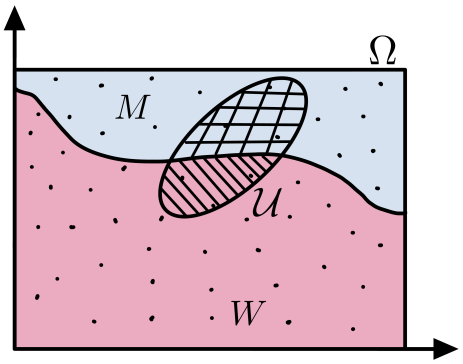

Figure 1: Прообразы всех возможных наблюдений эксперимента. ($\cdot$ – объект из X_N, M - man, W - women, $\mathcal{U} - \mathrm{g}^{-1}(x)$)

Зададимся вопросом, что требуется для полученной задачи ML, чтобы соответствовать вероятностной парадигме: фактически существование всех требуемых вероятностей:

- Пусть функции g_i измеримы.
- Пусть выборка (X_N, K_N) описывает всю генеральную совокупность объектов.
- **Крайне сильное предположение:** $\exists\, \mathbb{P}(X, K) \Rightarrow \exists\, \mathbb{P}(x)$, $\mathbb{P}(k)$, $\mathbb{P}(x \,|\, k)$, $\mathbb{P}(k \,|\, x)$
- Для каждой тройки $\{q(x), x, k\}$ определена функция потери $\mathcal{L} : K, X, K \to \mathbb{R}$
- $R(q) = \mathbb{E}_{X,K}\left[\mathcal{L}(q(x), x, k)\right] \longrightarrow \min\limits_{q \in M}$

$$\mathbb{E}_{X,K}\left[\mathcal{L}(q(x), k, x)\right] = \int\limits_{\Omega(X)} \int\limits_{\Omega(K)} p(x,k)\mathcal{L}(q(x), x, k) = \int\limits_{\Omega(X)} p(x) \int\limits_{\Omega(K)} p(k \,|\, x)\mathcal{L}(q(x), x, k)$$

Данная задача называется генеративной задачей машинного обучения, лежащей в основе Байесовского метода распознавания и используемая всюду, от *Sklearn* до глубокого обучения. Основная причина, почему вероятностная модель (говоря простыми словами, модель оценивающая меры множеств) настолько удобна заключается в том, что мы способны автоматически обрабатывать неточности данных, задавать аналитически плоскость лосса и прочие приятные фишки аппарата теории вероятностей.

3 Проблемы генеративных моделей

Огромное преимущество генеративных моделей заключается в том, что они обучают совместные распределения, что позволяет не только решать задачи нахождения K, но и генерировать X. Однако, несмотря на кажущееся удобство и силу генеративных моделей, их применение в реальных задачах часто сопровождается рядом характерных для подхода проблем: мы считаем себя в праве искать $\mathbb{P}(x), \mathbb{P}(k)$, что в действительности редкая роскошь. В особенности это кажется излишеством, когда от нас не требует понимать топологию X.

1. **Пример Не Байесовской задачи:** Пусть вы работаете на военной базе главным аналитиком. Вам поручено разработать систему "Свой-чужой", основанную на визуальных и ЭРЛС данных о самолетах противника и союзников. Имея опыт работы с данными вы быстро

понимаете, что речь идет о вероятностной задаче бинарной классификации. Однако можно ли применить в данном случае генеративный подход?

Оказывается, что нет! Первым же делом возникает вопрос: что такое априорная вероятность события "Вражеский самолет"? Как мы можем оценить вероятность события, что в действительности не может происходить в мирное время, а если и произошло, то это явно признак "неординарной" ситуации. Более формально ситуацию может разъяснить теория вероятностей: мы считаем вероятность события равной $p \iff$ при бесконечном проведении независимых опытов, событие будет воспроизводиться с частотой p. Соответсвенно, в данной задаче такой вероятности быть просто не может в мирное время, в котором система действовать и обязана.

2. **Пример неслучайной выборки:** Однако не всегда проблема кроется исключительно в задаче. Пусть задача состоит в оценке состояния оборудования завода по численным признакам с датчиков. Из огромного числа логов заказчик выбирает 100 наиболее характерных примеров состояний и таргет значений. Тогда такая ситуация так же становится не разрешимой с точки зрения генеративных моделей, так как неслучайная выборка не позволяет нам оценить априорные распределения событий.

Тогда перед исследователями встает вопрос, как поступить:

1. Отказаться от решения задачи оговоренным способом вероятностной парадигмы ML и оставить ее на откуп других специалистам.
2. Сказать "окей, мы не можем приблизить пространство $X \Rightarrow \mathbb{E}_k \left[\tilde{\mathcal{L}}(q(x), x, k) \,|\, X\right] \to \min\limits_{q \in M}$"

Последний выбор приводит к дискриминативной задаче машинного обучения, по наблюдению x предсказать k, лишенной проблемы генеративной и требующей много меньшего объема обучающих данных. Именно эту задачу на всем протяжении семестра вы изучаете.

4 Постановка задачи классификации

Пусть определена дискриминативная задача бинарной классификации. Обучающая выборка X, таргеты K, стратегия q. Далее мы сузим класс стратегий линейными моделями, однако сейчас никаких предположений на нее нет. Как определить дискриминативный лосс такой модели? Простейший способ сделать это – посчитать accuracy:

$$\mathcal{R}(q) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(q(x_i), x_i, k_i) = \frac{1}{N}\sum_{i=1}^{N} [q(x_i) \neq k_i]$$

Такой лосс является максимально простым и интуитивно понятным способом задать обучение классификатора q. Однако он сопровождается рядом проблем:

1. Такой лосс не является дифференцируемым в каноническом смысле $\Rightarrow$ оптимизация не более чем методами 0-порядка
2. (Доказательство рисунком) Поверхность данного лосса очень сложная с "постоянными" локальными минимумами. (рис. 3)

Попробуем приблизить данный лосс семейством функций (например рис. 2) сверху через отступ:

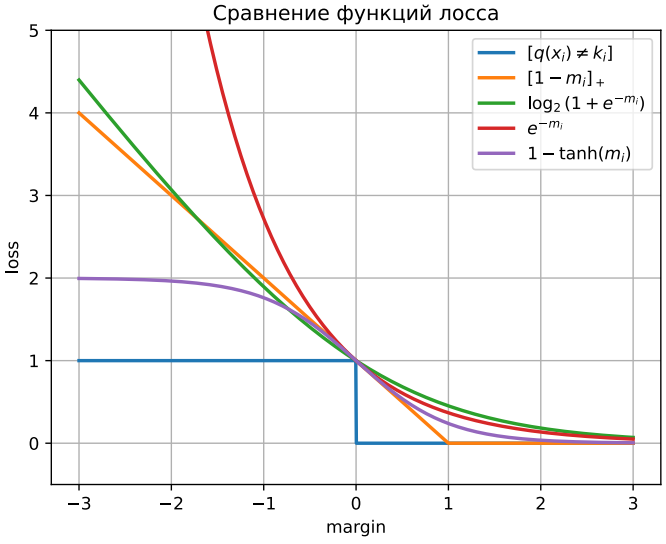

Figure 2: Примеры лосс функций

$$\mathcal{R}(q) = \frac{1}{N}\sum_{i=1}^{N} [q(x_i) \neq k_i] \leq \frac{1}{N}\sum_{i=1}^{N} \tilde{\mathcal{L}}(m_i) = \tilde{\mathcal{R}}(q) \Longrightarrow \tilde{\mathcal{R}}(q) \to \min \Longrightarrow \mathcal{R}(q) \to \min$$

Относительно подхода оценки лосса "в лоб" каждая такая функция выполняет свою роль на ура. Однако совсем не очевиден тот факт, что каждый лосс приведет к вероятностной стратегии.

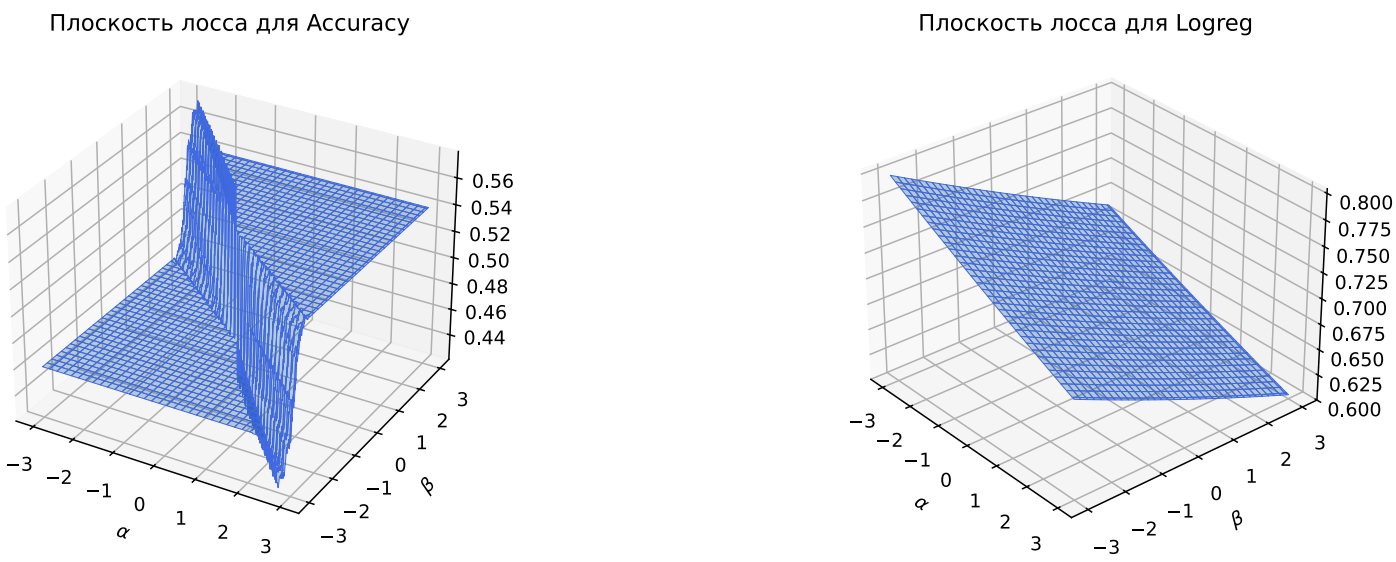

Figure 3: Демонстрация loss функций в пространстве параметров

5 Критерий вероятностной природы лосса

Пример: Пусть поставлена задача рекомендации пользователю рекламы. Аналитики компании уже проанализировали систему Ω и построена функция $g : \Omega \to \{1, 2, 3, \dots l\}$, что показывает некоторый категориальный признак для пары "реклама $\times$ пользователь". По полученной выборке (X_N, K_N) обучить стратегию определять вероятность того, что пользователь перейдет по рекламе.

$$\text{Тогда очевидно : } \sum_{i=1}^{N} \frac{\mathbb{1}\{Click|x_i = j\})}{\#(x = j)} = \frac{\text{кликов при } j}{\text{всего наблюдений } j} \xrightarrow[n\to\infty]{} p(Click \,|\, j)$$

Будем считать, что обучающая выборка нашей модели настолько велика, насколько мы захотим $(N \to \infty)$. Тогда для любого наблюдаемого значения $j \in \{1, 2, 3, \dots l\}$ мы сможем получить вероятность $p(\text{пользователь кликнул} \,|\, j) = 1 - p(\text{пользователь не кликнул} \,|\, j)$ и соответственно будем хотеть от нашей вероятностной модели выдавать ровно такие же значения на наблюдения x:

$$q(x = j) = p(Click|j)$$

Тогда понятно, как обобщить подобную потребность:

Критерий вероятностной природы лосс функции: (в качестве фукнции потерь крайне удобно брать фукнцию зависящую от марджина m_i)

$$arg\min_{q \in M} \tilde{\mathcal{R}}(q) = arg\min_{q \in M} \mathbb{E}\left[\tilde{\mathcal{L}}(q(x), x, k) = \tilde{\mathcal{L}}(m_i) \mid x\right] \xrightarrow[n\to\infty]{} p(k = +1 \,|\, \cdot)$$

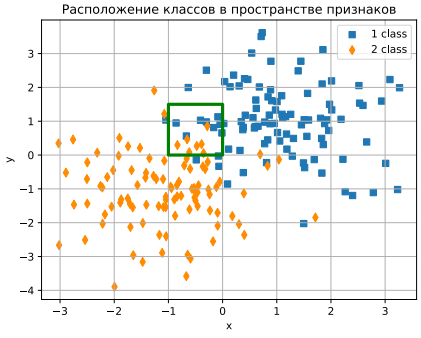

Figure 4: Квантизованная область пространства признаков с попавшими в нее объектами классов

5.1 Интерпретация данного результата на более сложных моделях

Пусть для пространства признаков выполнено условие "континуальности" (непрерывности из лекций Воронцова). Тогда мы фактически говорим, что отображение g ставит близкие объекты относительно изначального Ω близко друг другу и в пространстве признаков. Тогда мы можем говорить, что при его разбитии на малые области (**квантизация** X), разумная модель для каждого объекта будет выдавать вероятности, равные отношению классов в данной области (рис. 4).

$$\rho(x,c) \le \delta \Rightarrow \mathcal{R} = \sum \tilde{\mathcal{L}}(x) \approx \sum \left[\tilde{\mathcal{L}}(c) + \bar{\mathcal{O}}(x-c)\right] \approx \sum \tilde{\mathcal{L}}(c)$$

Функция потерь на близких объектах будет практически равна одному и тому же, а значит и при обучении на таких объектах должны в «идеале» выдаваться почти одинаковые вероятности.

5.2 Примеры проверки по критерию

Задача 1: Задача бинарной классификации: $\mathcal{L}(q,x,k) = (q(x) - [k=+1 \,|\, x])^2$ – MSE

$$arg\min_q \mathbb{E}(\mathcal{L}(q,x,k) \,|\, x) = arg\min_q \left[(q(x)-1)^2 \underbrace{p(k=+1|x)}_{p} + q^2(x) \underbrace{p(k=-1|x)}_{1-p} \right]$$

$$\frac{\partial}{\partial q(x)} \mathbb{E}(\mathcal{L}(q,x,k) \,|\, x) = 2(q(x)-1)p + 2q(x)(1-p) = 0 \Rightarrow q(x) = p(k=+1 \,|\, x)$$

Задача 2: Задача бинарной классификации: $\mathcal{L}(q,x,k) = |q(x) - [k=+1 \,|\, x]|$ – MAE

$$arg\min_q \mathbb{E}(\mathcal{L}(q,x,k) \,|\, x) = arg\min_q \left[|q(x)-1| \underbrace{p(k=+1|x)}_{p} + |q(x)| \underbrace{p(k=-1|x)}_{1-p} \right]$$

$$arg\min_q \Big[p(1-q(x)) + q(x)(1-p) \Big] = arg\min_q \left[p + q(x) - 2pq(x) \right] \Rightarrow q(x) \neq p(k=+1 \,|\, x)$$

6 Задание функции потерь из вероятностного распределения

Ранее рассматривался общий вид классификаторов. Теперь сузим семейство моделей до линейных классификаторов. Напомним их вид: $q(x \,|\, w) = f(\langle x, w \rangle)$, – где функция f осуществляет нормировку score на вероятность. Тогда будем считать модель порождения данных следующей:

1. Выбирается объект x из генеральной совокупности объектов
2. Объект x порождает распределение $p(k \,|\, x)$ над пространством таргет переменных по соответствующему закону.
3. Функция риска будет принимать вид Оценки правдоподобия

$$\mathcal{R} = \sum_{i=1}^{N} \log p(k_i \,|\, x_i) \Longrightarrow \exp\{\mathcal{R}\} = \prod_{i=1}^{N} p(k_i \,|\, x_i)$$

Достаточно заметить, что оценка правдоподобия является состоятельной, асимптотически эффективной и асимптотически нормальной $\Rightarrow$ для выводимых таким образом функций риска и лосс функций проверять критерий вероятностной природы не нужно (что очевидно из построения).

6.1 Logreg или великая Логистическая регрессия

Будем считать, что вероятность $p(k \,|\, x)$ для бинарной классификации есть Бернулевское параметрическое семейство $Bern(p)$ $\Rightarrow$ по объекту предсказываем p и его класс есть реализация Бернулевской случайной величины.

$$L = \exp\{\mathcal{R}\} = \prod_{i=1}^{N} q(x_i)^{[k_i=+1]} (1-q(x_i))^{[k_i=-1]} \to \max$$

$$-\log L = -\mathcal{R} = -\sum_{i=1}^{N} \{[k_i=+1] \log q(x_i) + [k_i=-1] \log(1-q(x_i))\} \to \min$$

Для вывода нашей модели воспользуемся подходом GLM – Generalized Linear Machine, что позволит получить точный вид для зависимости $q(x_i)$. Опуская точные выкладки и

дополнительную теорию на пол страницы мы узнаем: фукнция связи для распределения есть $\sigma(\theta) = \frac{1}{1+\exp\{-\theta\}}$. Теперь же просто подставим это в выражение для риска $\mathcal{R}$ и упростим выражение, считая, что θ есть параметр, моделируемый нашей линейной моделью: $\theta = \langle x, w\rangle$

$$-\log L = -\mathcal{R} = -\sum_{i=1}^{N} \{[k_i = +1] \log \sigma(\langle x, w\rangle) + [k_i = -1] \log(1 - \sigma(\langle x, w\rangle))\}$$

Далее заметим, что для сигмоиды выполняется: $1 - \sigma(\theta) = 1 - \frac{1}{1+e^{-\theta}} = \frac{1}{1+e^{\theta}} = \sigma(-\theta)$

$$\Rightarrow -\sum_{i=1}^{N} \{[k_i = +1] \log \sigma(\langle x, w\rangle) + [k_i = -1] \log \sigma(-\langle x, w\rangle)\} = -\sum_{i=1}^{N} \log \sigma(k_i \langle x, w\rangle)$$

$$\text{раскрывая сигмоиду} \Rightarrow -\log L = -\mathcal{R} = \sum_{i=1}^{N} \log\{1 + \exp\{-k_i \langle x, w\rangle\}\}$$

7 Оценка качества восстановления вероятностей модели

Построение вероятностной задачи не является панацеей, так как модель быть переобучена или недообученна на задачу, выбранное распределение таргет переменных может отличаться от наших ожиданий, тем самым вся модель будет предсказывать совершенно неестественные распределения на объектах. Хотелось бы вывести (аналитически) более эмпирическую оценку качества вероятностной модели, нежели «суррогатный» эмпирический риск $\mathcal{R}$.

Рассмотрим такую ситуацию: для задачи бинарной классификации определим набор моделей, каждая обучаемая на различные гипотезы об апостериорном распределении таргета и с различными гиперпараметрами. Тогда для каждой модели строим "диаграммы калибровки":

1. Разбиваем ось $[0, 1]$ на бины измельчением $\{a_1, a_2, \dots a_\ell\}$, $[a_i, a_{i+1}] = \mathbb{B}_i$
2. Для каждого объекта высчитываем оценку вероятности $+1$ класса $p_i \in \mathbb{B}_i$
3. Для каждого бина считаем отношение попавших в него действительно $+1$ классов к общему числу попавших.

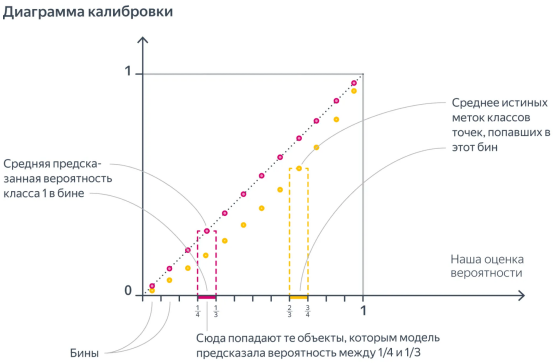

Figure 5: Диаграмма калибровки

Такие диаграммы выполняют вполне осязаемую функцию: показать, насколько сильно предсказанная вероятность модели отличается от действительных выборочных.
На примере (рис. 6) отчетливо видны 5 ситуаций:

1. **Desired:** Идеальное поведение классификатора, к которому мы стремимся при обучении вероятностной модели
2. **Uncertain on bound:** модель, что отдает сильное предпочтение сильно отдаленным объектам. В таком случае крайне вероятно переобучение.
3. **To confident:** модель предсказывает каждый из классов слишком уверенно
4. **patalogic +1 bias:** модель старательно отдает предпочтение $+1$ классу. Эта ситуация является поводом задуматься о смене либо модели данных, либо о классе моделей мл
5. **patalogic -1 bias:** противоположная ситуация предыдущей.

Примеры калибровочных диаграмм

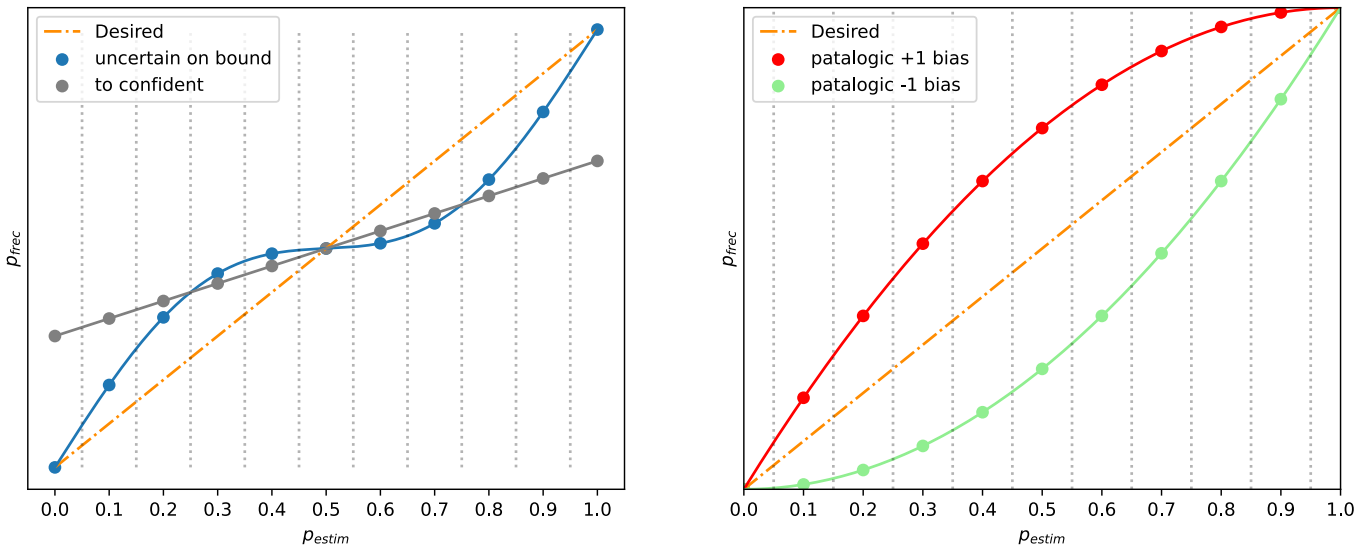

Figure 6: Примеры калибровачных диаграмм

7.1 Функционал качества оценки вероятности

В случае поведения вашей модели по сценарию **patalogic +1 bias** или **patalogic -1 bias** скорее всего вашему решению мало что поможет, кроме как смены архитектуры или рассматриваемой модели данных. Однако в остальных случаях есть шанс поднастроить модель, "просмотреть паттерн ее ошибок и вручную изменить ей выходные оценки". Семейство таких методов называется «калибровками». Для этого введем функционал качества калибровки **Expected/Maximum calibration error** по аналогии с диаграммой:

$$\mathbf{ECE} = \sum_{i=1}^{\ell} \frac{|\mathbb{B}_i|}{N} \left| \frac{1}{|\mathbb{B}_i|} \sum_{x \in \mathbb{B}_i} q(x) - \frac{1}{|\mathbb{B}_i|} \sum_{x \in \mathbb{B}_i} \mathbb{1}_{\{y_i = +1\}} \right|$$

$$\mathbf{MCE} = \max_{i \in \overline{1,\ell}} \frac{|\mathbb{B}_i|}{N} \left| \frac{1}{|\mathbb{B}_i|} \sum_{x \in \mathbb{B}_i} q(x) - \frac{1}{|\mathbb{B}_i|} \sum_{x \in \mathbb{B}_i} \mathbb{1}_{\{y_i = +1\}} \right|$$

Это лишь метрика качества, но не функционал для оптимизации, так как изменение модели влечет изменение и правой части, из-за чего его использование ограничивается оценкой качества.

7.2 Методы калибровки вероятностей

Методы калибровки предлагают не изменять модель напрямую. Напротив, они фиксируют веса модели, в то же время используя ее выход как новый признак объектов *(самый элементарный стакинг моделей, но это на будущее и для уже смешариков)*, что в теории позволяет усложнить модель и сделать ее более "нелинейной". Более того, в таком случае мы получаем возможность сделать вероятностную модель из невероятностной. Всего существует 3 наиболее применимых и популярных подхода:

- **Калибровка Платта** – самая очевидная и широко применяемая, основанная на "логистической регрессии" поверх выхода классификатора.
- **Isotonic regression** – классическая задача восстановления функции распределения поверх стратегии q через изотоническую регрессию.
- **Гистограммная калибровка** – упрощенный вариант изотонической регрессии, с фиксированными границами $\mathbb{B}_i$

7.2.1 Калибровка Платта

Изначально разрабатываемая для получения вероятности из марджина неверояностных моделей (в частности **SVM**), представляет собой по сути применение еще одного линейного слоя (линейной модели) от одного аргумента с применением сигмоиды. Обучаемыми параметрами являются α и β – масштаб и сдвиг:

$$p(k_i = +1 \,|\, x_i) = \sigma\left(\alpha q(x_i) + \beta\right) = \frac{1}{1 + \exp\{-\alpha q(x_i) + \beta\}}$$

$$\Longrightarrow \sum_{i=1}^{N} \log\left(1 + \exp\{-\alpha q(x_i) + \beta\}\right) \longrightarrow \min_{\alpha,\beta \in \mathbb{R}}$$

Усложнение модели Платта: вместо сигмоиды можно использовать какую-нибудь другую функцию, дающую на выходе число от 0 до 1, например функцию плотности бета распределения (рис. 7):

$$p(k = +1 \,|\, q(x), \alpha_1, \beta_1) = \mathcal{B}(q(x), \alpha_1, \beta_1) = \frac{q(x)^{\alpha_1 - 1}\left(1 - q(x)\right)^{\beta_1 - 1}}{\mathcal{B}(\alpha_1, \beta_1)}$$

$$p(k = -1 \,|\, q(x), \alpha_2, \beta_2) = \mathcal{B}(q(x), \alpha_2, \beta_2) = \frac{q(x)^{\alpha_2 - 1}\left(1 - q(x)\right)^{\beta_2 - 1}}{\mathcal{B}(\alpha_2, \beta_2)}$$

Руководствуясь леммой Неймана-Пирсона, найдем отношение плотностей вероятности и обучим параметры на максимум правдоподобия:

$$LR = \frac{q(x)^{\alpha_1 - 1}\left(1 - q(x)\right)^{\beta_1 - 1}}{q(x)^{\alpha_2 - 1}\left(1 - q(x)\right)^{\beta_2 - 1}} \cdot \frac{\mathcal{B}(\alpha_2, \beta_2)}{\mathcal{B}(\alpha_1, \beta_1)} = \frac{q(x)^a}{\left(1 - q(x)\right)^b} K \simeq \frac{q(x)^a}{\left(1 - q(x)\right)^b} e^{-c}$$

Здесь мы приблизили отношение Бета функций экспонентой. Теперь же превратим отношение правдоподобия в вероятность, $a \geq 0$, $b \geq 0$, и получим классический вид **Бета калибровки**:

$$p(k = +1 \,|\, q(x), a, b, c) = \frac{1}{1 + LR^{-1}} = \frac{1}{1 + e^c \,/\, \frac{q(x)^a}{(1 - q(x))^b}}$$

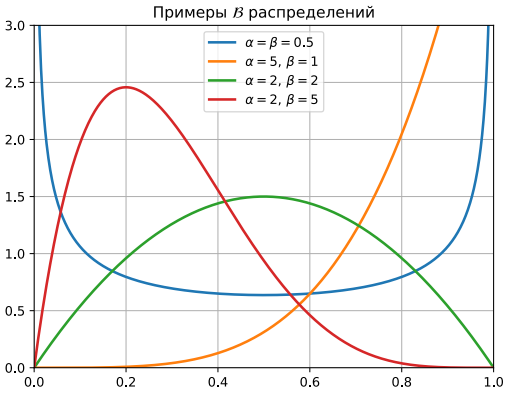

Figure 7: Примеры $\mathcal{B}$ распределений

7.2.2 Калибровка Isotonic regression

Строится монотонно неубывающая кусочно-постоянная функция деформации оценок алгоритма (рис. 8). Требуется восстановить функцию $g(q(x))$, параметрами которой является измельчение сегмента $[0, 1]$, на каждом сегменте измельчения $g(t) = \texttt{const}$, $g(t_1) \leq g(t_2)$, $t_1 \leq t_2$, что в общем случае решается специальными солверами (например pool adjacent violators algorithm за $O(n)$).

$$\sum_{i=1}^{N} \left(k_i - g(q(x_i))\right)^2 \longrightarrow \min_{g}$$

7.2.3 Гистограмная калибровка

Является частным случаем Изотонической при фиксированных равных интервалах $\mathbb{B}_i$. Такой метод предсказывает лишь конечное число вероятностей, являющееся гиперпараметром алгоритма.

$$\sum_{i=1}^{\ell} \left| \frac{1}{\mathbb{B}_i} \sum_{j=1}^{N} \mathbb{1}_{\{q(x_i) \in \mathbb{B}_i\}} k_j - \theta_i \right| \longrightarrow \min_{\vec{\theta}}$$

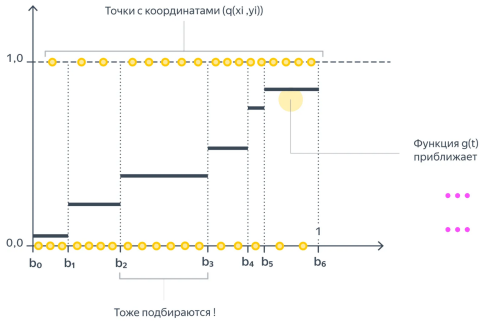

Figure 8: Определение Изотонической калибровки

8 GLM

Теперь тонкий лед: вы будете проходить всю дальнейшую теорию на 4 курсе на Байесовских методах машинного обучения, поэтому все дальнейшие рассуждения будут носить лишь ознакомительный характер.

Модель *GLM* уже встречалась вам на лекциях Воронцова, ее суть заключается в описании семейтсва вероятностных стратеги через использование распределений **экспоненциального семейства** (с распределениями этого семейства каждый из вас встречался на курсе Теории вероятности и Математической статистики). В сущности, исследователь исходя из своих первичных знаний о задаче строит гипотезу о том, как выглядит распределение $p(k \mid x)$ (в частности как была разобрана logreg, там было Бернулевское распределение), после чего выводит зависимости параметров распределения от наблюдения.

Общий вид экспоненциального распределения: $p(x \mid \theta) = \frac{f(x)}{g(\theta)} \exp\{\theta^T \vec{u}(x)\}$. Далее окажется, что наши модели стремятся приблизить именно θ – вектор параметров распределения по наблюдению. Однако сейчас мы рассмотрим частное представление семейства, что должно быть вам знакомо:

$$p(k_i \mid \theta_i, \varphi_i) = \exp\left\{\frac{k_i\theta_i - c(\theta_i)}{\varphi_i} + h(\theta_i, \varphi_i)\right\}$$

Оказывается, что $\mu_i = \mathbb{E}k_i = c'(\theta_i) \Longrightarrow \theta_i = (c')^{-1}(\mathbb{E}k_i) = \mathrm{g}(\mu_i); \mathbb{D}k_i = \varphi_i c''(\theta_i)$ данное выражение в полной мере вы пощупаете в следующем году, нужно лишь сказать, что данное выражение является частным от много более общей теории, с которой в данном курсе мы не встречаемся.

8.1 Бернулевское распределение

$$p(k_i \mid \mu_i) = \mu^{k_i}(1-\mu_i)^{1-k_i} = \{\text{приведем к явному виду}\} = \exp\left\{k_i \underbrace{\ln\frac{\mu_i}{1-\mu_i}}_{\theta_i = logit} + \ln(1-\mu_i)\right\}$$

Значит параметр θ_i есть совсем не сама вероятность μ_i, а некоторая функция от нее: $\theta_i = \mathrm{g}(\mu_i) = \ln\frac{\mu_i}{1-\mu_i}$. Следовательно $\mu_i = \mathrm{g}^{-1}(\theta_i) = \frac{1}{1+e^{\theta_i}} = \sigma(\theta_i)$! Таким образом логистическая регрессия восстанавливает θ по которой мы уже и находим параметр вероятности.

8.2 Гауссовское распределение

$$p(k_i \mid \mu_i, \sigma_i^2) = \frac{1}{\sqrt{2\pi}\sigma_i}\exp\left\{-\frac{(k_i-\mu_i)^2}{2\sigma_i^2}\right\} = \exp\left\{\frac{k_i\mu_i + -0.5\mu_i^2}{\sigma_i^2} - \frac{k_i^2}{2\sigma_i^2} - 0.5\ln(2\pi\sigma_i^2)\right\}$$

$$\Longrightarrow \theta_i = \mu_i, \quad \varphi_i = \sigma_i^2$$

Получив формулы оценок на парамтеры строим модели, что по объекту предсказывают θ_i и φ_i и подставляем в функционал эмпирического риска для оптимизации.

1. Десять лекций по статистическому и структурному распознаванию образов. Михаил Иванович Шлезингер, Вацлав Главач
2. https://education.yandex.ru/handbook/ml/article/kak-ocenivat-veroyatnosti
3. Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration

Специальное спасибо Алексееву Илье за помощь с редакцией конспекта