Методы классификации

Зинина Анастасия

Линейный классификатор в задаче бинарной классификации

- $Y = \{+1, -1\}$
- Модель алгоритмов $a(x, \alpha) = sign(\alpha_0 + \sum_{j=1}^{n} \alpha_j f_j(x)) = sign < \alpha, x >$, где α — вектор параметров.
- Если $a(x, \alpha) > 0$, то алгоритм a относит объект x к классу +1, иначе к классу -1.
- Уравнение $a(x, \alpha) = 0$ описывает разделяющую поверхность.
- Величина $M_i = y_i < \alpha, x_i >$ называется отступом (margin) i-го объекта относительно алгоритма классификации
- Если $M_i < 0$, то алгоритм $a(x, \alpha)$ допускает ошибку на i-ом объекте .
- Чем больше отступ M_i, тем правильнее и надёжнее классификация объекта.

Аппроксимация эмпирического риска

- Естественный функционал качества - доля неправильных ответов

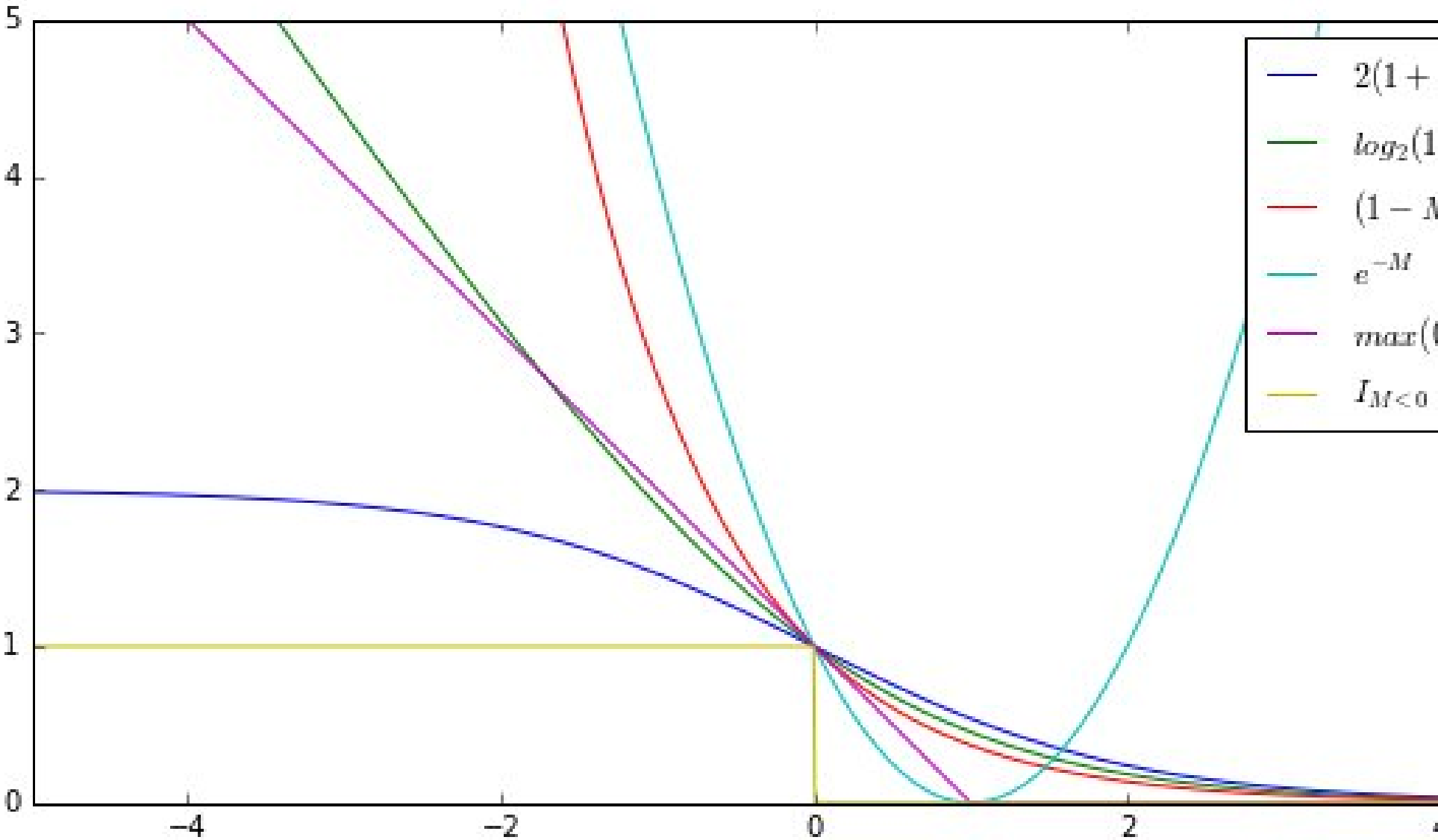
$$Q(a, l) = \frac{1}{l} \sum_{i=1}^{l} [a(x_i) \neq y_i] = \frac{1}{l} \sum_{i=1}^{l} I_{M_i<0}$$

- Рассмотрим $L(M_i)$ — монотонно невозрастающую функция отступа, мажорирующая пороговую функцию потерь: $I_{M_i<0} < L(M_i)$.

- Тогда минимизацию суммы таких функций отступа можно рассматривать как приближённый метод минимизации числа ошибок на обучающей выборке:

$$Q(a, x) = \sum_{i=1}^{l} I_{M_i<0} \leqslant \tilde{Q}(a, x^l) = \sum_{i=1}^{l} L(M_i)$$

Часто используемые функции потерь L(M)



Линейный дискриминант Фишера

- Находим веса, минимизируя квадратичную функцию потерь:

$$Q(\alpha, X) = \frac{1}{l}\sum_{i=1}^{l}(y_i - < \alpha, x_i >)^2 \to \min_{\alpha}$$

- А что, если хотим вычислить не просто метки, а вероятности принадлежности объекта классу?

Логистическая регрессия

- $Y = \{0, 1\}$
- Хотим предсказать $P(y = 1 \mid x) \equiv \pi(x)$
- $\pi(x) = 1 \cdot P(y = 1 \mid x) + 0 \cdot P(y = 0 \mid x) = E(y \mid x)$
- Однако находить $\pi(x) \approx < w, x >$ нельзя, т.к. может не выполняться $< \alpha, x > \in [0, 1]$
- Тогда нужна функция $g : (0, 1) \to [0, 1]$, для восстановления которой мы можем использовать линейную регрессию:

$$g(E(y \mid x)) \approx < \alpha, x >$$

- После нахождения оценки $g(E(y \mid x))$ можем получить оценку для $E(y \mid x)$
- Используем $\pi(x) \approx g^{-1}(< \alpha, x >) = \frac{1}{1+e^{-<\alpha, x>}}$
- Тогда $\underbrace{\ln \frac{\pi(x)}{1 - \pi(x)}}_{\text{логит}} \approx < \alpha, x >$

Настройка параметров

- $L(\alpha, X) = \prod p(x_i, y_i) = \prod P(y_i \mid x_i) p(x_i) \to \max\limits_{\alpha}$ эквивалентно $\prod P(y_i \mid x_i) \to \max\limits_{\alpha}$
- $P(y_i \mid x_i) = P(y_i = 1 \mid x_i)^{y_i} P(y_i = 0 \mid x_i)^{1-y_i} = \pi(x_i)^{y_i} (1 - \pi(x_i))^{1-y_i}$
- $L(w, X) \to max$ эквивалентно $-\ln L(\alpha, X) \to min$:

$$-\ln L(\alpha, X) = -\sum_{i=1}^{l} (y_i \ln \pi(x_i) + (1 - y_i) \ln(1 - \pi(x_i)) \to min$$

- Этот функционал назван log-loss

Настройка параметров

- Если перейти к обозначениям $Y = \{-1, 1\}$, то получим

$$\frac{P(+1 \mid x)}{P(-1 \mid x)} = e^{<\alpha, x>}$$

$$P(y = +1 \mid x) = \frac{1}{1 + e^{-<\alpha, x>}} = \sigma(< \alpha, x >)$$

$$P(y = -1 \mid x) = \frac{1}{1 + e^{<\alpha, x>}} = \sigma(- < \alpha, x >),$$

  где $\sigma(z) = \frac{1}{1+e^{-z}}$

- Тогда можем записать

$$P(y_i \mid x_i) = \sigma(y_i < \alpha, x_i >)$$

- И настраиваем параметры с помощью ОМП приходим к логарифмической функции потерь:

$$-\ln L(\alpha, X) = -\sum_{i=1}^{l} \ln \frac{1}{1 + e^{-y_i <\alpha, x_i>}} = \sum_{i=1}^{l} \ln(1 + e^{-y_i <\alpha, x_i>})$$

One vs. all classification

- Y принимает N значений
- Для отделения каждого класса i(обозначим его 1) от остальных (обозначим как 0) обучаем классификатор $a^{(i)}(x) = P(y = 1 \mid x)$
- Для нового объекта выбираем $i = \underset{i}{argmax}\ a^{(i)}(x)$

Обобщенные линейные модели

Перейдём к конструированию обобщённых линейных моделей, частными случаями которых были линейная и логистическая регрессии.

Экспоненциальное семейство

Распределение является экспоненциальным, если его можно представить в виде

$$p(y;\gamma) = b(y)exp(\eta^T T(y) - a(\eta)),$$

где функции $T(y)$,$a(y)$, $b(y)$ задают параметрическое семейство,а η является параметром.

Структура обобщенных линейныех моделей

Обобщённые линейные модели (Generalized Linear Models, GLM) состоят из трёх частей:

- Случайная компонента, определяющая условное распределение y при данном x. И это распределение принадлежит экспоненциальному семейству с параметром η
- Линейная функция признаков

$$\eta = \alpha_0 + \alpha_1 f_1(x) + ... + \alpha_n f_n(x)$$

- Гладкая функция связи, которая переводит условное ожидание $\mu = E(y \mid x)$ в линейную функцию:

$$g(\mu) = \eta = \alpha_0 + \alpha_1 f_1(x) + ... + \alpha_n f_n(x)$$

Примеры GML

Линейная регрессия с МНК:

- $y \mid x \sim N(\mu, \sigma^2)$ - нормальное распределение принадлежит экспонентному семейству и имеет параметр $\eta = \mu$
- $\eta = \alpha^T x$
- В качестве функции связи возьмём тождественную

$$g(\mu) = \mu = \eta = \alpha^T x$$

Логистическая регрессия:

- $y \mid x \sim Bernoulli(p)$ - нормальное распределение принадлежит экспонентному семейству и имеет параметр $\eta = \ln \frac{p}{1-p}$
- $\eta = \alpha^T x$
- В качестве функции связи возьмём логит (помним, что $p = P(y = 1 \mid x) = E(y \mid x)$)

$$g(\mu) = \ln \frac{\mu}{1-\mu} = \eta = \alpha^T x$$

Обобщение логистической регрессии на случай многоклассовой классификации

Теперь аналогично сконструируем алгоритм softmax-регрессии

Softmax-регрессия

- $Y = \{1, 2, ..., k\}$
- $p(y = i; \varphi) = \varphi_i$, $p(y = k; \varphi) = 1 - \sum_{i=1}^{k-1} \varphi_i$
- $y \mid x \sim multinomial(\varphi)$
- Определим вектор T(y) размерности k-1 так, что у вектора T(k) k-ая координата равна 1, остальные 0: $(T(y))_k = I_{y=k}$
- Покажем, что мультиномиальное распределение принадлежит экспоненциальному семейству:

$$p(y; \varphi) = \varphi_1^{I_{y=1}} \varphi_2^{I_{y=2}} ... \varphi_k^{I_{y=k}}$$

$$= \varphi_1^{(T(y))_1} \varphi_2^{(T(y))_2} ... \varphi_k^{1-\sum (T(y))_k}$$

$$= exp((T(y))_1 \log(\varphi_1/\varphi_k) + (T(y))_2 \log(\varphi_2/\varphi_k) +$$

$$... + (T(y))_{k-1} \log(\varphi_{k-1}/\varphi_k) + \log(\varphi_k))$$

$$= b(y) exp(\gamma^T T(y) - a(\gamma))$$

Softmax-регрессия

- Здесь параметр определяется как $\eta_i = \log \frac{\varphi_i}{\varphi_k}$
- Определим $\eta_k = \log \frac{\varphi_k}{\varphi_k} = 0$
- Получим выражение для вероятности принадлежности у классу i ϕ_i:

$$e^{\eta_i} = \frac{\varphi_i}{\varphi_k}$$

$$\varphi_k e^{\eta_i} = \varphi_i$$

$$\varphi_k \sum_{i=1}^{k} e^{\eta_i} = \sum_{i=1}^{k} \varphi_i = 1$$

$$\varphi_i = \frac{e^{\eta_i}}{\sum_{j=1}^{k} e^{\eta_j}}$$ - функция softmax

Softmax-регрессия

- $\eta_i = \alpha_i^T x$ для $i = 1, ..., k-1$, $\eta_k = \alpha_k^T x = 0$, где $\alpha_i \in R^{n+1}$ - наши параметры
- Выразим вероятности ϕ_i:

$$p(y = i \mid x) = \varphi_i = \frac{e^{\eta_i}}{\sum\limits_{j=1}^{k} e^{\eta_j}} = \frac{e^{\alpha_i^T x}}{\sum\limits_{j=1}^{k} e^{\alpha_j^T x}}$$

- Настройка параметров - ОМП:

$$L(\alpha) = \sum_{i=1}^{l} \log \prod_{k=1}^{n} \left(\frac{e^{\alpha_i^T x_i}}{\sum\limits_{j=1}^{n} e^{\alpha_j^T x_i}} \right)^{I_{\{y^{(i)}=k\}}} \to max$$

Метрики качества в задачах классификации

- Очевидной мерой качества в задаче классификации является доля правильных ответов (accuracy):

$$accuracy = \frac{\sum_{i=1}^{l} I_{a(x_i)=y_i}}{l}$$

- Этого может быть недостаточно: в случае несбалансированных классов может быть выгоднее причислять все объекты к мажорантному классу. Т.е. один из классов не распознается, а доля правильных ответов высока.

  NB Базовая доля — доля правильных ответов алгоритма, всегда выдающего наиболее мощный класс.

- Для сравнения алгоритмов a_1 и a_2 с долями правильных ответов r_1 и r_2 соответственно, причем $r_2 > r_1$, используем относительное уменьшением ошибки алгоритма a_2 называется величина $\frac{(1-r_1)-(1-r_2)}{1-r_1}$

Матрица ошибок

	y=1	y=0
a(x)=1	TP	FP
a(x)=0	FN	TN

$$precision = \frac{TP}{TP+FP},\ recall = \frac{TP}{TP+FN}$$

- Можно регулировать точность и полноту, изменяя порог t в классификаторе $a(x) = I_{b(x)>t}$.
- Объединим точность и полноту в одну метрику: F-мера, гармоническое среднее точности и полноты:

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

- Можно использовать R-точность(breakeven point).
  Она вычисляется как точность при таком t, при котором полнота равна точности:

$$R - precision = precision(I_{b(x)>t^*}),$$

$$t^* = \underset{t}{argmin} \mid precision(I_{b(x)>t}) - recall(I_{b(x)>t}) \mid$$

AUC-ROC

- Как мы видели на примере логистической регрессии, алгоритмы бинарной классификации могут быть устроены так: $a(x) = I_{b(x)>t}$, где t - пороговое значение.
- Если алгоритм работает плохо, может быть непонятно: плох сам алгоритм или неправильно выбран порог.
- Если мы хотим оценить алгоритм до установления порога, то рассмотрим новую метрику.

AUC-ROC

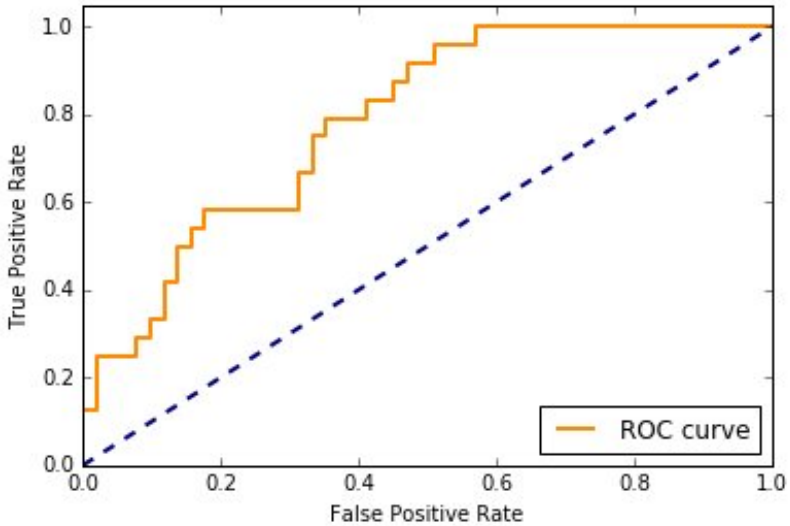
- Рассмотрим двумерное пространство, одна из координат которого FPR, а другая — TPR. Откладываем точки на графике, соответствующие разным значениям порога.

$$FPR = \frac{FP}{FP + TN}; \quad TPR = \frac{TP}{TP + FN}$$

- Всего различных значений порога l+1
- Отсортируем выборку по значению $b(x_i)$
- Если в качестве порога выберем $\max\limits_i b(x_i)$, то получим точку (0,0).
- Если в качестве порога выберем $\min\limits_i b(x_i) - \varepsilon$, то получим точку (1,1).

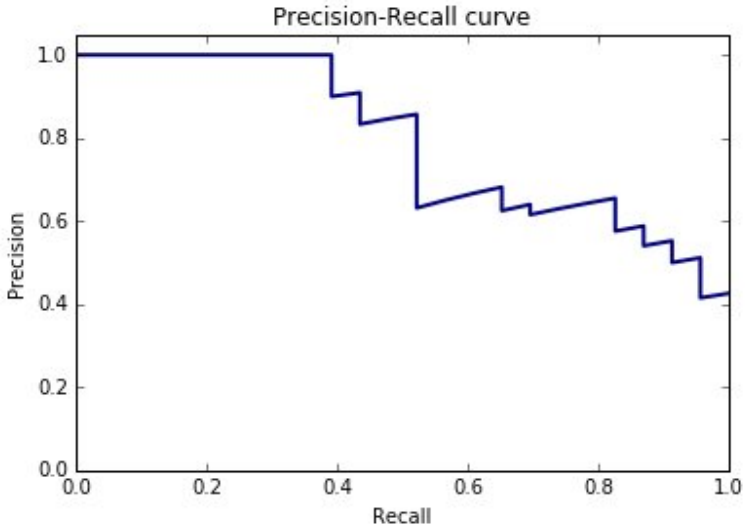
AUC-ROC

- Если текущий объект относится к классу «1», то у алгоритма увеличивается TPR. Тогда ROC-кривая сдвигается вверх на $\frac{1}{l_1}$(где l_1 — число объектов класса "1").
- Если у текущего объекта класс «0», то ROC-кривая сдвигается вправо на $\frac{1}{l_0}$ (l_0 — число объектов класса "0").
- В качестве метрики рассмотрим площадь под ROC-кривой (Area Under ROC Curve, AUC-ROC).
- Чем больше значение AUC-ROC, тем лучше: если можно выбрать порог так, чтобы классификатор не делал ошибок, то значение AUC-ROC будет равно 1.



AUC-PR

- По осям откладываются полнота (по оси абсцисс) и точность (по оси ординат).
- Идём по ранжированной выборке.
- Если объект относится к классу «0», то полнота не меняется, точность падает, кривая опускается вниз.
- Если же объект относится к классу «1», то полнота увеличивается на $\frac{1}{l_1}$, точность растет, и кривая поднимается вправо и вверх.



Несбалансированные классы

- Рассмотрим случай несбалансированных классов.
- Пусть у нас есть выборка размером 1 000 100, из которых 100 относятся к классу "1".
- Рассмотрим 2 алгоритма:
  - алгоритм 1: относит к классу ”1” 100 документов, 90 из них - правильно;
  - алгоритм 2: относит к классу ”1” 2000 документов, 90 из них - правильно;
- Алгоритмы находят одинаковое число объектов класса ”1”, но второй возвращает много False positive. Однако в случае с AUC-ROC разница между алгоритмами будет небольшая из-за преобладания в выборке класса "0".
  - алгоритм 1: 0,9 TPR, 0,00001 FPR;
  - алгоритм 2: 0,9 TPR, 0,00191 FPR.
- Если рассмотреть AUC-PR, то разница между алгоритмами будет более заметна.
  - алгоритм 1: 0,9 recall, 0,9 precision;
  - алгоритм 2: 0,9 recall, 0,045 precision.

Использование метрик

- Посмотрим на формулы для precision/recall и FPR/TPR:

$$TPR = \frac{TP}{TP + FN}, FPR = \frac{FP}{FP + TN}$$

$$precision = \frac{TP}{TP + FP}, recall = \frac{TP}{TP + FN}$$

- AUC-PR концентрируется на классе ”1”,в то время как AUC-ROC учитывет и то, как классифицируется класс "0".
- Тогда можно рекомендовать следующее: если важно классифицировать хорошо оба класса, то использовать стоит AUC-ROC (например, при классификации картинок "кошки vs собаки"). Если же важен только класс "1"(например, медицинская диагностика), то можно использовать AUC-PR.

Список литературы

- *Andrew Ng* CS229 Lecture notes
  http://cs229.stanford.edu/notes/cs229-notes1.pdf
- *McCullagh and Nelder* Generalized Linear Models - CHAPMAN AND HALL,1989