

Методы машинного обучения. Предобработка данных и оценивание моделей

Воронцов Константин Вячеславович

www.MachineLearning.ru/wiki?title=User:Vokov

вопросы к лектору: k.vorontsov@iai.msu.ru

материалы курса:

github.com/MSU-ML-COURSE/ML-COURSE-24-25

орг.вопросы по курсу: ml.cmc@mail.ru

1 Предварительная обработка данных

- Преобразование признаков
- Обработка пропущенных значений
- Генерация признаков

2 Оценки качества классификации

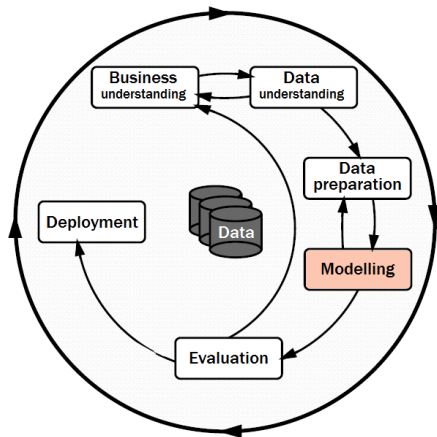
- Чувствительность, специфичность, ROC, AUC
- Правдоподобие вероятностной модели классификации
- Точность, полнота, AUC-PR

3 Анализ ошибок и выбор моделей

- Обобщающая способность
- Анализ ошибок
- Выбор моделей

Межотраслевой стандарт интеллектуального анализа данных

CRISP-DM: Cross Industry Standard
Process for Data Mining (1999)



Компании-инициаторы:

- SPSS
- Teradata
- Daimler AG
- NCR Corp.
- OHRA

Шаги процесса:

- понимание бизнеса
- понимание данных
- **предобработка данных**
- **моделирование**
- **оценивание моделей**
- внедрение

Шкалы измерения

Измерительная шкала — множество Z допустимых значений, получаемых в результате измерения признака $f(x)$, $f: X \rightarrow Z$

Тип шкалы определяется множествами

- допустимых биективных преобразований $\psi: Z \rightarrow Z'$
- допустимых операций над значениями из шкалы Z

Классификация типов измерительных шкал по Стивенсу:

| шкала | D | $\psi(z)$ | операции |
|-------------------------|--------------|------------|-----------------------|
| логическая (boolean) | 0, 1 | биективные | $\vee \wedge \neg$ |
| номинальная (nominal) | $< \infty$ | биективные | $= \neq \in$ |
| порядковая (ordinal) | $< \infty$ | монотонные | $= \neq \in < >$ |
| интервальная (interval) | \mathbb{R} | $az + b$ | $< > + -$ |
| отношений (ratio) | \mathbb{R} | az | $< > + - \times \div$ |
| абсолютная (absolute) | \mathbb{R} | z | любые |

S.S.Stevens. On the Theory of Scales of Measurement // Science, 1946.

Примеры величин, измеряемых в различных шкалах

- **Логическая**
наличие/отсутствие свойства, ответ «да/нет»
- **Номинальная** (можно переименовать или перенумеровать)
идентификаторы классов, людей, регионов, фирм, товаров
- **Порядковая** (порядок частичный или линейный)
уровень образования, тяжесть болезни, степень согласия
- **Ранговая** (частный случай порядковой: $1, 2, 3, \dots, N$)
оценка в баллах, шкалы Рихтера, Бофорта, Мооса, Бека
- **Интервальная** (можно сдвигать положение нуля)
время, географическая широта, температура ($^{\circ}\text{C}$, $^{\circ}\text{F}$)
- **Отношений** (можно менять единицы измерения)
масса, скорость, объём, сила, давление, заряд, яркость, $^{\circ}\text{K}$
- **Абсолютная**
число предметов, частота события, оценка вероятности

Ослабление шкалы

Номинальный \rightarrow много бинарных (one-hot-encoding):

- $f_v(x) = [f(x) = v]$, для всех значений v признака
- $f_A(x) = [f(x) \in A]$, индикаторный признак подмножества A

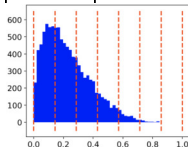
Числовой или порядковый \rightarrow бинарный:

- $f_{a,b}(x) = [a \leq f(x) \leq b]$ для заданного отрезка $[a, b]$

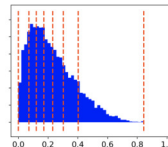
Числовой \rightarrow ранговый (data binning, quantization):

- $f_a(x) = \sum_{k=1}^K [f(x) \geq a_k]$, номер интервала сетки a_1, \dots, a_K

равномерная сетка



квантильная сетка



Ослабление шкалы всегда влечёт потерю информации

Усиление шкалы

Номинальный \rightarrow числовой:

- категория заменяется частотой:

$$f'(x) = \frac{1}{\ell} \sum_{i=1}^{\ell} [f(x_i) = f(x)]$$

- условное среднее числового признака $g(x)$:

$$f'(x) = \text{mean}(g|f(x)) = \frac{\sum_{i=1}^{\ell} g(x_i) [f(x_i) = f(x)]}{\sum_{i=1}^{\ell} [f(x_i) = f(x)]},$$

- условное среднее целевой величины $y(x)$:

$$f'(x) = \text{mean}(y|f(x)), \text{ возможно переобучение!}$$

Порядковый \rightarrow числовой:

- значение заменяется частотой:

$$f'(x) = \frac{1}{\ell} \sum_{i=1}^{\ell} [f(x_i) \leq f(x)]$$

Нормализация и стандартизация числовых шкал

Многие методы накапливают меньше вычислительных погрешностей, если признаки приведены к одному масштабу

- $f'_j(x) = \frac{f_j(x) - f_j^{\min}}{f_j^{\max} - f_j^{\min}}$ — нормализация, приведение к $[0, 1]$
- $f'_j(x) = \frac{f_j(x)}{|f_j|^{\max}}$ — масштабирование с сохранением нуля
- $f'_j(x) = \frac{f_j(x) - \mu_j}{\sigma_j}$ — стандартизация

f_j^{\max} , $|f_j|^{\max}$, f_j^{\min} , μ_j , σ_j определяются по обучающей выборке

Для повышения устойчивости к выбросам можно отбрасывать 5% наименьших и наибольших значений признака

Трансформация вида распределения

F_j — функция распределения (с.d.f.) признака f_j

Эмпирическая функция распределения (кусочно-постоянная):

$$\hat{F}_j(z) = \frac{1}{\ell} \sum_{i=1}^{\ell} [f_j(x_i) \leq z]$$

- $f'_j(x) = F_j(f_j(x))$ — преобразование $f_j(x)$ в равномерную на отрезке $[0, 1]$ случайную величину
- $f'_j(x) = \Phi^{-1}(F_j(f_j(x)))$ — преобразование $f_j(x)$ в случайную величину с заданной функцией распределения Φ (например, в нормальную)
- $f'_j(x) = \ln(1 + f_j(x))$ — преобразование неотрицательной случайной величины «с тяжёлым правым хвостом» (объёмы производства, перевозок, продаж)

Подходы к обработке пропущенных значений

- Игнорировать объекты или признаки с пропусками
— ведёт к потере информации :(
- Заполнить пропущенные значения признака f :
— средним или медианным значением \bar{f}
- Прогнозировать значения признака f по остальным:
— регрессия для вещественного признака f
— классификация для дискретного признака f
— матричные разложения, например, разреженный SVD
- Использовать модели, способные обрабатывать пропуски:
— решающие деревья
— голосование низкоразмерных базовых предикторов
- Ввести бинарный признак $f'(x) = [f(x) \text{ не известно}]$

Непараметрическая регрессия для заполнения пропусков

Формула Надарая–Ватсона, ядерное сглаживание:

$$\hat{f}_j(x_i) = \frac{\sum_u f_j(u) S(u, x_i)}{\sum_u S(u, x_i)}$$

где \sum_u — сумма по всем объектам $u \in X^\ell$ с известным $f_j(u)$

Возможные конструкции функций сходства $S(u, x)$:

- $S(u, x) = K\left(\frac{\rho(u, x)}{h}\right)$, $\rho^2(u, x) = \frac{1}{|J_{ux}|} \sum_{j \in J_{ux}} (f_j(u) - f_j(x))^2$
- $S(u, x) = \frac{1}{|J_{ux}|} \sum_{j \in J_{ux}} f_j(u) f_j(x)$ — скалярное произведение
- $S(u, x) = \frac{\sum_{j \in J_{ux}} f_j(u) f_j(x)}{\sqrt{\sum_{j \in J_{ux}} f_j^2(u)} \sqrt{\sum_{j \in J_{ux}} f_j^2(x)}}$ — косинусная ф.сх.

где J_{ux} — множество признаков j с известными $f_j(x)$ и $f_j(u)$

Разреженное низкоранговое матричное разложение

Дано: матрица $F = (f_{ij} = f_j(x_i))_{\ell \times n}$, $\Omega \subseteq \{1, \dots, \ell\} \times \{1, \dots, n\}$

Найти: матрицы $G = (g_{it})_{\ell \times k}$ и $U = (u_{jt})_{n \times k}$ такие, что

$$\|F - GU^T\| = \sum_{(i,j) \in \Omega} \underbrace{(f_{ij} - \langle g_i, u_j \rangle)_{\varepsilon_{ij}}^2}_{\varepsilon_{ij}} = \sum_{(i,j) \in \Omega} \left(f_{ij} - \sum_{t=1}^k g_{it} u_{jt} \right)^2 \rightarrow \min_{G, U}$$

Классический SVD неприменим для разреженной задачи.

Метод стохастического градиента: перебираем $(i, j) \in \Omega$ в случайном порядке, делаем градиентные шаги $(\varepsilon_{ij})^2 \rightarrow \min_{g_i, u_j}$

$$g_{it} := g_{it} + \eta \varepsilon_{ij} u_{jt}, \quad t = 1, \dots, k$$

$$u_{jt} := u_{jt} + \eta \varepsilon_{ij} g_{it}, \quad t = 1, \dots, k$$

$\hat{f}_j(x_i) = \langle g_i, u_j \rangle$ — восстановление пропущенных значений

g_{it} — новые признаки x_i в пространстве размерности k

Классические подходы к генерации признаков

Feature Engineering: признаки вычисляются по формулам, которые зависят от задачи, требуют изобретательности и знаний предметной области. Долго, дорого.

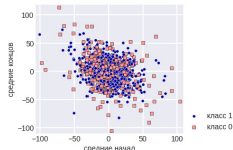
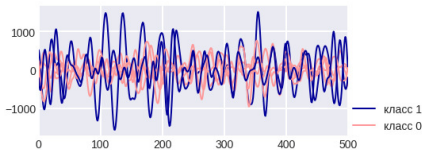
- Прогнозирование временных рядов:
признаки агрегируются по предыстории различной глубины
- Распознавание лиц:
признаки размера и формы черт лица
- Классификация и поиск текстов:
признаки частоты слов, терминов, названий, синонимов
- Распознавание речи:
спектральные, фонетические, лингвистические признаки

Иногда удачные признаки решают задачу без ML

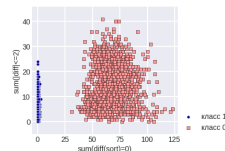
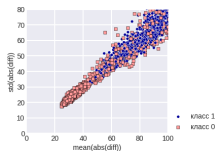
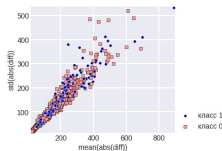
Соревнование «Ford Classification Challenge» (2008)

Задача детектирования поломок по сигналу датчика

Признаки, генерируемые по исходным временным рядам, слабы:



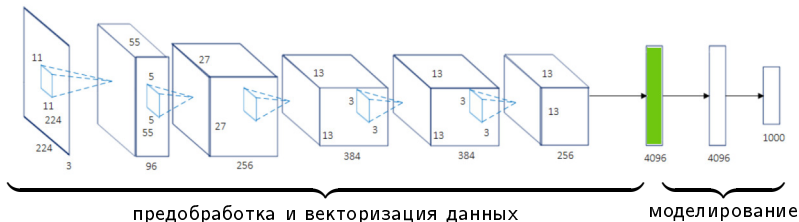
Среди признаков рядов их производных оказывается идеальный:



<https://dyakonov.org/2018/06/28/простые-методы-анализа-данных>

Обучаемая векторизация данных

Глубокие нейронные сети объединяют два этапа обработки данных: векторизацию и предсказательное моделирование



- компьютерное зрение
- обработка текстов естественного языка
- анализ сигналов и прогнозирование временных рядов
- распознавание и синтез речи
- анализ графов и транзакционных данных

— немного в следующем семестре, много в курсе DL

Анализ ошибок классификации

Задача классификации на два класса, $y_i \in \{-1, +1\}$.

Алгоритм классификации $a(x_i) \in \{-1, +1\}$

| | ответ классификатора | правильный ответ |
|--------------------|----------------------|------------------|
| TP, True Positive | $a(x_i) = +1$ | $y_i = +1$ |
| TN, True Negative | $a(x_i) = -1$ | $y_i = -1$ |
| FP, False Positive | $a(x_i) = +1$ | $y_i = -1$ |
| FN, False Negative | $a(x_i) = -1$ | $y_i = +1$ |

Правильность классификации (чем больше, тем лучше):

$$\text{Accuracy} = \frac{1}{\ell} \sum_{i=1}^{\ell} [a(x_i) = y_i] = \frac{\text{TP} + \text{TN}}{\text{FP} + \text{FN} + \text{TP} + \text{TN}}$$

Недостаток: не учитывается ни численность (дисбаланс) классов, ни цена ошибки на объектах разных классов.

Функции потерь, зависящие от штрафов за ошибку

Задача классификации на два класса, $y_i \in \{-1, +1\}$.

Модель классификации: $a(x; w, w_0) = \text{sign}(g(x, w) - w_0)$.

Чем больше w_0 , тем больше x_i таких, что $a(x_i) = -1$.

Пусть λ_y — штраф за ошибку на объекте класса y .

Функция потерь теперь зависит от штрафов:

$$\mathcal{L}(w, x_i) = \lambda_{y_i} [a(x_i; w, w_0) \neq y_i] = \lambda_{y_i} [(g(x_i, w) - w_0)y_i < 0].$$

Проблема

На практике штрафы $\{\lambda_y\}$ могут пересматриваться

- Нужен удобный способ выбора w_0 в зависимости от $\{\lambda_y\}$, не требующий построения w заново.
- Нужна характеристика качества модели $g(x, w)$, не зависящая от штрафов $\{\lambda_y\}$ и численности классов.

Определение ROC-кривой

Кривая ошибок ROC (receiver operating characteristic).

Каждая точка кривой соответствует некоторому $a(x; w, w_0)$.

- по оси X : доля ошибочных положительных классификаций (FPR — false positive rate):

$$\text{FPR} = \frac{\sum_{i=1}^{\ell} [y_i = -1][a(x_i; w, w_0) = +1]}{\sum_{i=1}^{\ell} [y_i = -1]};$$

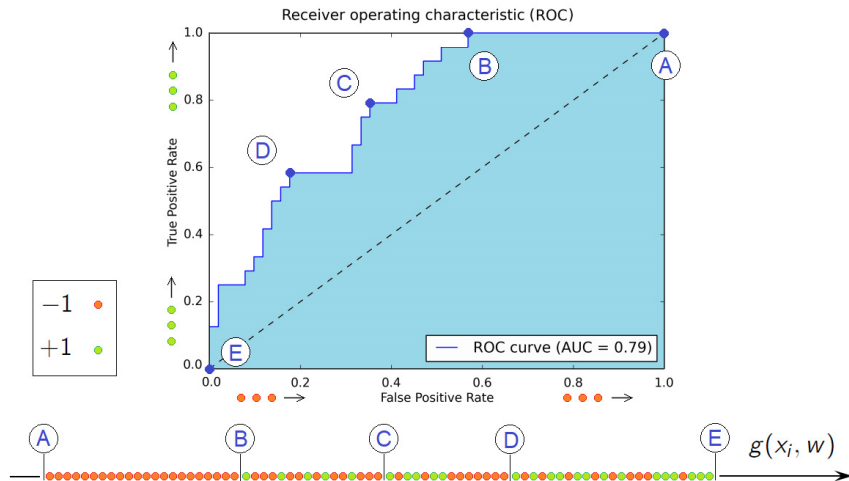
$1 - \text{FPR}$ называется специфичностью алгоритма a .

- по оси Y : доля правильных положительных классификаций (TPR — true positive rate):

$$\text{TPR} = \frac{\sum_{i=1}^{\ell} [y_i = +1][a(x_i; w, w_0) = +1]}{\sum_{i=1}^{\ell} [y_i = +1]};$$

TPR называется также чувствительностью алгоритма a .

ROC-кривая и площадь под кривой AUC (Area Under Curve)



ABCDE — положения порога w_0 на оси значений функции g

Алгоритм эффективного построения ROC-кривой

Вход: выборка $\{x_i\}_{i=1}^{\ell}$; дискриминантная функция $g(x, w)$;

Выход: ROC-кривая $(X_j, Y_j)_{j=0}^k$, $k \leq \ell$ и площадь AUC

$\ell_y := \sum_{i=1}^{\ell} [y_i = y]$, для всех $y \in Y$;

упорядочить $\{x_i\}$ по убыванию $g_i = g(x_i, w)$: $g_1 \geq \dots \geq g_{\ell}$;

$(X_0, Y_0) := (0, 0)$; $AUC := 0$; $\Delta X := 0$; $\Delta Y := 0$; $j := 1$;

для $i := 1, \dots, \ell$

$\Delta X := \Delta X + \frac{1}{\ell_-} [y_i = -1]$;

$\Delta Y := \Delta Y + \frac{1}{\ell_+} [y_i = +1]$;

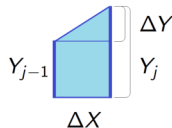
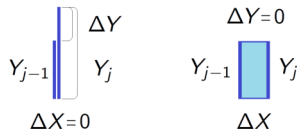
если $(g_i \neq g_{i-1})$ **то**

$X_j := X_{j-1} + \Delta X$;

$Y_j := Y_{j-1} + \Delta Y$;

$AUC := AUC + \frac{1}{2} (Y_{j-1} + Y_j) \Delta X$;

$j := j + 1$; $\Delta X := 0$; $\Delta Y := 0$;



Максимизация AUC как попарного (pairwise) критерия

Модель классификации: $a(x_i, w, w_0) = \text{sign}(g(x_i, w) - w_0)$.

AUC — это доля правильно упорядоченных пар (x_i, x_j) :

$$\begin{aligned} \text{AUC}(w) &= \frac{1}{\ell_-} \sum_{i=1}^{\ell} [y_i = -1] \text{TPR}_i = \\ &= \frac{1}{\ell_- \ell_+} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} [y_i < y_j] [g(x_i, w) < g(x_j, w)] \rightarrow \max_w \end{aligned}$$

Явная максимизация аппроксимированного AUC:

$$1 - \text{AUC}(w) \leq Q(w) = \sum_{i,j: y_i < y_j} \underbrace{L(g(x_j, w) - g(x_i, w))}_{M_{ij}(w)} \rightarrow \min_w$$

$L(M)$ — убывающая функция отступа,

$M_{ij}(w)$ — новое понятие отступа для пар объектов.

Алгоритм SG для максимизации AUC

Возьмём для простоты линейный классификатор:

$$g(x, w) = \langle x, w \rangle, \quad M_{ij}(w) = \langle x_j - x_i, w \rangle, \quad y_i < y_j.$$

Вход: выборка X^ℓ , темп обучения h , темп забывания λ ;

Выход: вектор весов w ;

инициализировать веса w_j , $j = 0, \dots, n$;

инициализировать оценку: $\bar{Q} := \frac{1}{\ell + \ell_-} \sum_{i,j} [y_i < y_j] L(M_{ij}(w))$;

повторять

выбрать **пару объектов** (i, j) : $y_i < y_j$, случайным образом;

вычислить потерю: $\varepsilon_{ij} := L(M_{ij}(w))$;

сделать градиентный шаг: $w := w - h L'(M_{ij}(w))(x_j - x_i)$;

оценить функционал: $\bar{Q} := (1 - \lambda)\bar{Q} + \lambda\varepsilon_{ij}$;

пока значение \bar{Q} и/или веса w не сойдутся;

Логарифм правдоподобия, log-loss

Вероятностная модель бинарной классификации, $y_i \in \{-1, +1\}$:

$$a(x, w) = \text{sign}(g(x, w) - w_0), \quad g(x, w) = P(y = +1|x, w).$$

Проблема: ROC и AUC инвариантны относительно монотонных преобразований дискриминантной функции $g(x, w)$.

Критерий логарифма правдоподобия (log-loss):

$$Q(w) = \sum_{i=1}^{\ell} [y_i = +1] \ln g(x, w) + [y_i = -1] \ln(1 - g(x, w)) \rightarrow \max_w$$

Вероятностная модель многоклассовой классификации:

$$a(x) = \arg \max_{y \in Y} P(y|x, w);$$

$$Q(w) = \sum_{i=1}^{\ell} \ln P(y_i|x_i, w) \rightarrow \max_w$$

Точность и полнота бинарной классификации

В информационном поиске:

$$\text{Точность, Precision} = \frac{TP}{TP+FP}$$

$$\text{Полнота, Recall} = \frac{TP}{TP+FN}$$

Precision — доля релевантных среди найденных

Recall — доля найденных среди релевантных

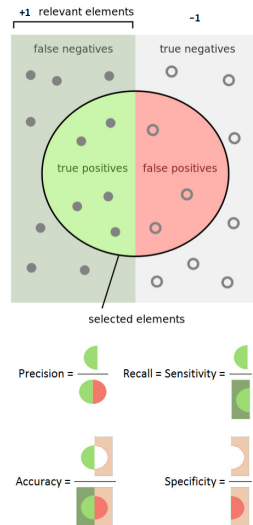
В медицинской диагностике:

$$\text{Чувствительность, Sensitivity} = \frac{TP}{TP+FN}$$

$$\text{Специфичность, Specificity} = \frac{TN}{TN+FP}$$

Sensitivity — доля верных положительных диагнозов

Specificity — доля верных отрицательных диагнозов



Точность и полнота многоклассовой классификации

Для каждого класса $y \in Y$:

TP_y — верные положительные

FP_y — ложные положительные

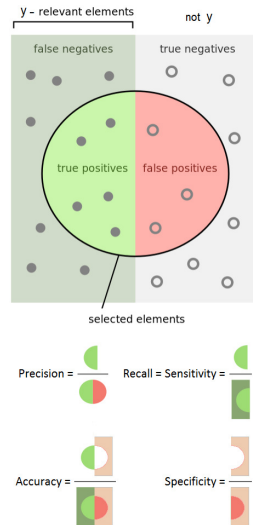
FN_y — ложные отрицательные

Точность и полнота с микроусреднением:

$$\text{Precision: } P = \frac{\sum_y TP_y}{\sum_y (TP_y + FP_y)};$$

$$\text{Recall: } R = \frac{\sum_y TP_y}{\sum_y (TP_y + FN_y)};$$

Микроусреднение не чувствительно
к ошибкам на малочисленных классах



Точность и полнота многоклассовой классификации

Для каждого класса $y \in Y$:

TP_y — верные положительные

FP_y — ложные положительные

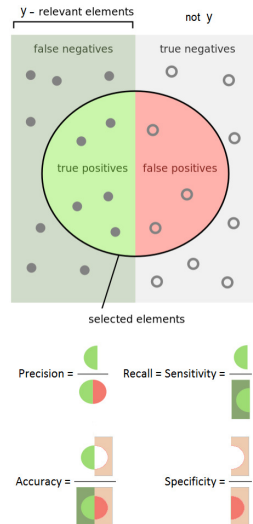
FN_y — ложные отрицательные

Точность и полнота **с макроусреднением**:

$$\text{Precision: } P = \frac{1}{|Y|} \sum_y \frac{TP_y}{TP_y + FP_y};$$

$$\text{Recall: } R = \frac{1}{|Y|} \sum_y \frac{TP_y}{TP_y + FN_y};$$

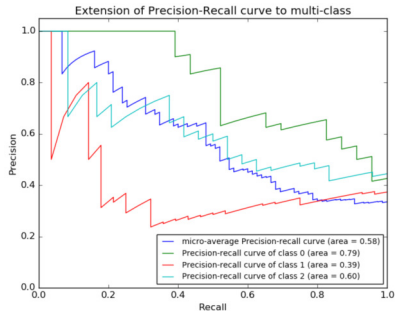
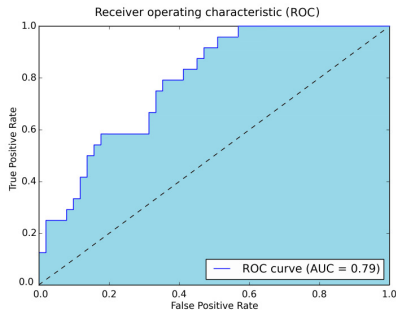
Макроусреднение чувствительно
к ошибкам на малочисленных классах



Кривые ROC и Precision-Recall

Модель классификации: $a(x) = \text{sign}(\langle x, w \rangle - w_0)$

Каждая точка кривой соответствует значению порога w_0



AUROC — площадь под ROC-кривой

AUPRC — площадь под кривой Precision-Recall

Примеры из Python scikit learn: <http://scikit-learn.org/dev>

Резюме. Оценки качества классификации

- Чувствительность и специфичность лучше подходят для задач с несбалансированными классами
- Логарифм правдоподобия (log-loss) лучше подходит для оценки качества вероятностной модели классификации.
- Точность и полнота лучше подходят для задач поиска, когда доля объектов релевантного класса очень мала.

Агрегированные оценки:

- AUC лучше подходит для оценивания качества, когда соотношение цены ошибок не фиксировано.
- AUPRC — площадь под кривой точность–полнота.
- $F_1 = \frac{2PR}{P+R}$ — F -мера, другой способ агрегирования P и R .
- $F_\beta = \frac{(1+\beta^2)PR}{\beta^2P+R}$ — F_β -мера: чем больше β , тем важнее R .

Задачи оценивания и выбора моделей

Дано: X — пространство объектов, Y — множество ответов

$X^\ell = (x_i, y_i)_{i=1}^\ell$ — обучающая выборка, $y_i = y^*(x_i)$

$A_t = \{a: X \times W_t \rightarrow Y\}$ — параметрические модели, $t \in T$

W_t — пространство параметров модели A_t

$\mu_t: (X \times Y)^\ell \rightarrow W_t$ — методы обучения, $t \in T$

Найти: метод μ_t с наилучшей *обобщающей способностью*.

Частные случаи:

- выбор лучшей модели A_t (model selection);
- выбор метода обучения μ_t для заданной модели A (в частности, оптимизация *гиперпараметров*);
- отбор признаков (feature selection):
 $F = \{f_j: X \rightarrow D_j: j = 1, \dots, n\}$ — множество признаков;
метод обучения μ_J использует только признаки $J \subseteq F$.

Обобщающая (предсказательная) способность метода

$\mathcal{L}(w, x)$ — функция потерь модели $a(w, x)$ на объекте x ;

$Q(w, X^\ell) = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathcal{L}(w, x_i)$ — функционал качества $a(w, x)$ на X^ℓ

Внутренний критерий оценивает качество на обучении X^ℓ :

$$Q_\mu(X^\ell) = Q(\mu(X^\ell), X^\ell).$$

Недостаток: эта оценка смещена, т.к. μ минимизирует её же.

Внешний критерий оценивает качество «вне обучения», например, по отложенной (hold-out) контрольной выборке X^k :

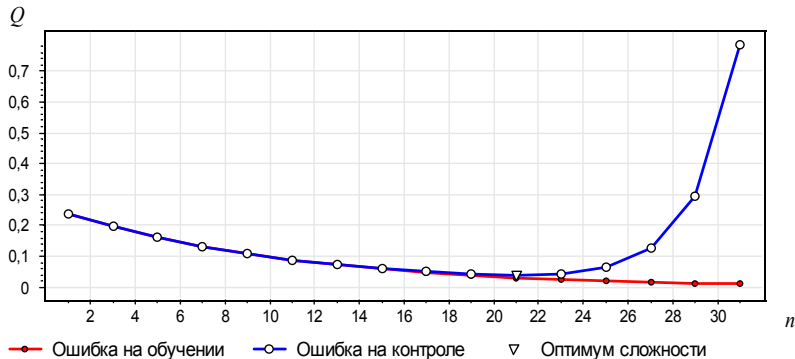
$$Q_\mu(X^\ell, X^k) = Q(\mu(X^\ell), X^k).$$

Недостаток: эта оценка зависит от разбиения $X^L = X^\ell \sqcup X^k$.

Основное отличие внешних критериев от внутренних

Внутренний критерий монотонно убывает с ростом сложности модели (например, числа признаков).

Внешний критерий имеет характерный минимум, соответствующий оптимальной сложности модели.



Кросс-проверка (cross-validation, CV)

Усреднение оценок hold-out по заданному N — множеству разбиений $X^L = X_n^\ell \sqcup X_n^k$, $n = 1, \dots, N$:

$$CV(\mu, X^L) = \frac{1}{|N|} \sum_{n \in N} Q_\mu(X_n^\ell, X_n^k).$$

Частные случаи — разные способы задания множества N .

1. $|N| = 1$ — единственное разбиение: hold-out.
2. N — случайное множество разбиений: метод Монте-Карло.
3. N — множество всех $C_{\ell+k}^k$ разбиений:
полная кросс-проверка (complete cross-validation, CCV).

Недостаток: оценка CCV вычислительно слишком сложна.
Используются либо малые k , либо комбинаторные оценки CCV.

Скольльзящий контроль и поблочная кросс-проверка

4. Скользящий контроль (leave one out CV): $k = 1$,

$$\text{LOO}(\mu, X^L) = \frac{1}{L} \sum_{i=1}^L Q_{\mu}(X^L \setminus \{x_i\}, \{x_i\}).$$

Недостатки LOO: ресурсоёмкость, высокая дисперсия.

5. Кросс-проверка по q блокам (q -fold CV): случайное разбиение $X^L = X_1^{\ell_1} \sqcup \dots \sqcup X_q^{\ell_q}$ на q блоков (почти) равной длины,

$$\text{CV}_q(\mu, X^L) = \frac{1}{q} \sum_{n=1}^q Q_{\mu}(X^L \setminus X_n^{\ell_n}, X_n^{\ell_n}).$$

Недостатки q -fold CV:

- оценка существенно зависит от разбиения на блоки;
- каждый объект лишь один раз участвует в контроле.

Множественная поблочная кросс-проверка

6. Контроль t раз по q блокам ($t \times q$ -fold CV)

— стандарт «де факто» для тестирования методов обучения.

Выборка X^L разбивается t раз случайным образом на q блоков

$$X^L = X_{s1}^{\ell_1} \sqcup \dots \sqcup X_{sq}^{\ell_q}, \quad s = 1, \dots, t, \quad \ell_1 + \dots + \ell_q = L;$$

$$CV_{t \times q}(\mu, X^L) = \frac{1}{t} \sum_{s=1}^t \frac{1}{q} \sum_{n=1}^q Q_{\mu}(X^L \setminus X_{sn}^{\ell_n}, X_{sn}^{\ell_n}).$$

Преимущества $t \times q$ -fold CV:

- увеличением t можно улучшать точность оценки (компромисс между точностью и временем вычислений);
- каждый объект участвует в контроле ровно t раз;
- оценивание доверительных интервалов (95% при $t = 40$).

Методология анализа ошибок

$\mathcal{L}(w, x_i)$ — функция потерь (чем меньше, тем лучше).

Критерий средней потери модели $a(x, w)$ на выборке U :

$$Q(w, U) = \frac{1}{|U|} \sum_{x_i \in U} \mathcal{L}(w, x_i)$$

Анализ потерь на обучающей выборке:

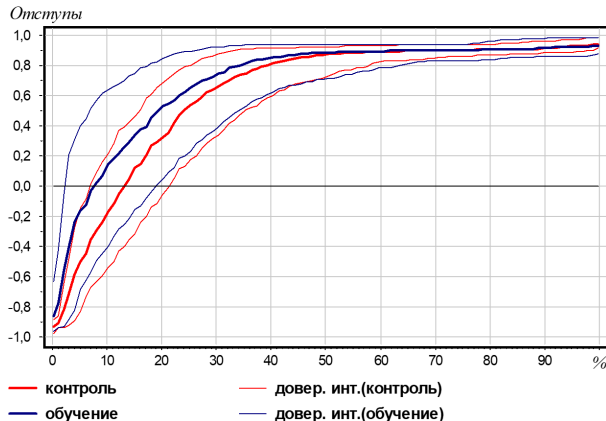
- Ранжировать объекты по убыванию потерь $\mathcal{L}_i = \mathcal{L}(x_i, w)$
- Объекты со сверхбольшими потерями — выбросы?
- Если нет, то как улучшить модель на этих объектах?

Сравнительный анализ потерь на обучении и тесте:

- Сильно ли отличаются распределения потерь?
- Если сильно, то как устранить переобучение?
- Объекты со сверхбольшими отличиями — выбросы?

Анализ распределения отступов в задаче классификации

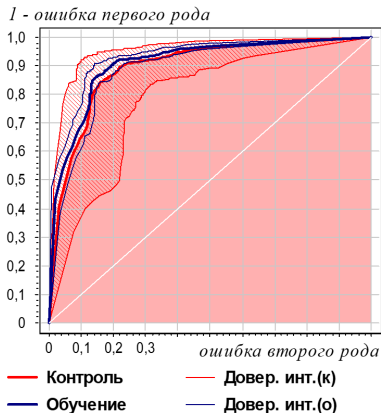
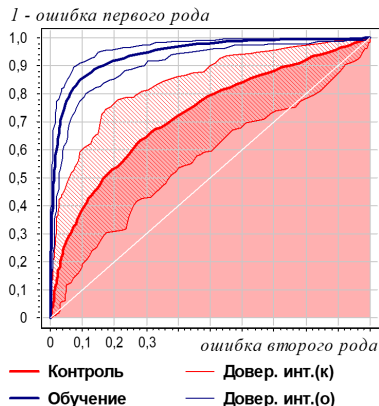
Вместо потерь $L(M_i)$ можно ранжировать отступы M_i



Задача UCI:australian, метод JRip

Анализ ROC-кривых

ROC-кривые можно строить отдельно для каждого класса



Задача UCI:liver, метод Bagging

Задача UCI:heart, метод Naive Bayes

A/B тестирование (A/B testing, Split Testing)

Две модели, «базовая A» и «улучшенная B», построенные по историческим данным X^ℓ , тестируются по метрике качества Q на новых данных X^k

В чём отличия A/B тестирования от обычного hold-out?

- X^k — это именно будущие данные (out-of-time), а не часть прошлых данных, исключённых из обучения (out-of-sample)
- больше реализма: за это время могут измениться свойства потока данных, реальные данные не обязаны быть i.i.d.
- однократный выбор модели почти не переобучается
- накопление данных X^k может потребовать много времени
- работа модели может влиять на формирование потока данных (например, в рекомендательных системах)

Мета-обучение (meta-learning, learning to learn)

Дано: выборка «задача, метод» → критерии качества

Найти: модель предсказания, каким методом решать задачу

Критерий: точность предсказания оптимального метода

Признаки:

- размерные характеристики задачи
- характеристики пространства признаков:
типы, выбросы, пропуски, корреляции
- результаты быстрых низкоразмерных методов

Joaquin Vanschoren. Meta-learning Architectures: Collecting, Organizing and Exploiting Meta-knowledge. 2009.

Joaquin Vanschoren. Meta-Learning: A Survey. 2018.

Автоматический выбор моделей и гиперпараметров (AutoML)

Проблема:

подбор структуры модели (архитектуры нейросети)
и гиперпараметров требует слишком много ресурсов

Дано: выборка «задача, структура» → критерии качества

Найти: какой следующий эксперимент провести с моделью

Критерий:

минимизация затрат ресурсов на автоматический поиск
оптимальной модели, сопоставимой по качеству с моделями,
построенными профессиональными исследователями

Близкая классическая задача — *планирование экспериментов*

Xin He et al. AutoML: A Survey of the State-of-the-Art. 2019

<https://github.com/sberbank-ai-lab/LightAutoML> — AutoML от Сбербанка

- Культура анализа данных:
 - смотреть на данные глазами
 - пробовать нетривиальные идеи предобработки, основанные на знаниях предметной области
 - использовать анализ ошибок и визуализацию
 - креативно порождать и оценивать больше гипотез
 - знать и учитывать сильные и слабые стороны методов
- Автоматизация распространяется по схеме CRISP-DM, в перспективе нас ожидает бесшовная интеграция этапов
 - предобработки данных
 - моделирования
 - оценивания и выбора моделей
 - внедрения