

Методы машинного обучения. Обзор оптимизационных задач

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

ВМК МГУ • 15 октября 2024

1 Обучение с учителем

- Классификация и регрессия
- Регуляризация
- Обучение ранжированию

2 Обучение без учителя

- Восстановление плотности распределения
- Кластеризация и частичное обучение
- Обучаемая векторизация объектов

3 Неклассические парадигмы обучения

- Перенос обучения и многозадачное обучение
- Обучение с привилегированной информацией
- Типология задач машинного обучения

Общая оптимизационная задача машинного обучения

Дано: выборка объектов $X^\ell = \{x_1, \dots, x_\ell\}$

Найти: вектор параметров w модели $a(x, w)$

Критерий: минимум эмпирического риска

$$Q(w, X^\ell) = \sum_{i=1}^{\ell} \mathcal{L}(w, x_i) \rightarrow \min_w$$

где $\mathcal{L}(w, x_i)$ — функция потерь модели $a(x, w)$ на объекте x_i ,
или минимум регуляризованного эмпирического риска

$$Q(w, X^\ell) = \sum_{i=1}^{\ell} \mathcal{L}(w, x_i) + \sum_{j=1}^r \tau_j R_j(w) \rightarrow \min_w$$

где R_j — регуляризаторы, τ_j — коэффициенты регуляризации

Напоминание. Задача обучения бинарной классификации

Обучающая выборка: $X^\ell = (x_i, y_i)_{i=1}^\ell$, $x_i \in \mathbb{R}^n$, $y_i \in \{-1, +1\}$

- ❶ Фиксируется модель классификации, например, *линейная*:

$$a(x, w) = \text{sign } g(x, w) = \text{sign } \sum_{j=1}^n w_j f_j(x) = \text{sign} \langle x, w \rangle$$

- ❷ Функция потерь — убывающая функция отступа $L(\text{margin})$

$$\mathcal{L}(w, x_i) = [g(x_i, w)y_i < 0] \leq L(g(x_i, w)y_i)$$

- ❸ Метод обучения — *минимизация эмпирического риска*:

$$Q(w, X^\ell) = \sum_{i=1}^\ell [g(x_i, w)y_i < 0] \leq \sum_{i=1}^\ell L(g(x_i, w)y_i) \rightarrow \min_w$$

- ❹ Проверка модели w^* по тестовой выборке $X^k = (\tilde{x}_i, \tilde{y}_i)_{i=1}^k$:

$$Q(w^*, X^k) = \frac{1}{k} \sum_{i=1}^k [g(\tilde{x}_i, w^*)\tilde{y}_i < 0]$$

Напоминание. Задача обучения многоклассовой классификации

Обучающая выборка: $X^\ell = (x_i, y_i)_{i=1}^\ell$, $x_i \in \mathbb{R}^n$, $y_i = y(x_i) \in Y$

- ❶ Модель классификации, например, *линейная*, $w = (w_y)_{y \in Y}$:

$$a(x, w) = \arg \max_{y \in Y} g(x, w_y) = \arg \max_{y \in Y} \langle x, w_y \rangle$$

- ❷ Функция потерь — бинарная или её аппроксимация:

$$\mathcal{L}(w, x_i) = \sum_{z \neq y_i} [g(x_i, w_{y_i}) < g(x_i, w_z)] \leq \sum_{z \neq y_i} L(M_{iz}(w)),$$

где $M_{iz}(w) = g(x_i, w_{y_i}) - g(x_i, w_z)$ — отступ x_i по классу z

- ❸ Метод обучения — минимизация эмпирического риска:

$$Q(w, X^\ell) = \sum_{i=1}^\ell \sum_{z \neq y_i} L(g(x_i, w_{y_i}) - g(x_i, w_z)) \rightarrow \min_w$$

- ❹ Проверка модели w^* по тестовой выборке $X^k = (\tilde{x}_i, \tilde{y}_i)_{i=1}^k$

Напоминание. Задача обучения регрессии

Обучающая выборка: $X^\ell = (x_i, y_i)_{i=1}^\ell$, $x_i \in \mathbb{R}^n$, $y_i = y(x_i) \in \mathbb{R}$

- ❶ Модель регрессии, например, *линейная*, $w \in \mathbb{R}^n$:

$$a(x, w) = \langle x, w \rangle = \sum_{j=1}^n w_j f_j(x)$$

- ❷ Функция потерь — *квадратичная*:

$$\mathcal{L}(w, x_i) = (a(x_i, w) - y_i)^2$$

- ❸ Метод обучения — *метод наименьших квадратов*:

$$Q(w, X^\ell) = \sum_{i=1}^\ell \mathcal{L}(w, x_i) = \sum_{i=1}^\ell (a(x_i, w) - y_i)^2 \rightarrow \min_w$$

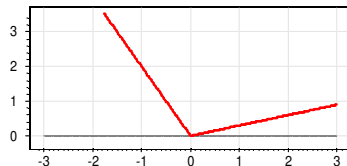
- ❹ Проверка модели w^* по тестовой выборке $X^k = (\tilde{x}_i, \tilde{y}_i)_{i=1}^k$:

$$Q(w^*, X^k) = \frac{1}{k} \sum_{i=1}^k (a(\tilde{x}_i, w^*) - \tilde{y}_i)^2$$

Квантильная регрессия

Функция потерь, $\varepsilon = a(x_i, w) - y_i$:

$$L(\varepsilon) = \begin{cases} C_+ |\varepsilon|, & \varepsilon > 0 \\ C_- |\varepsilon|, & \varepsilon < 0; \end{cases}$$



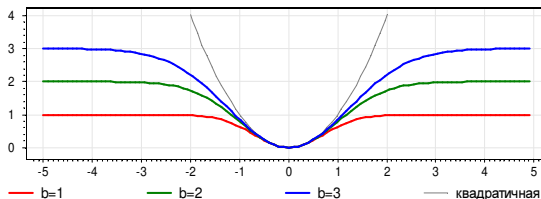
Модель $a(x, w) = w$: решением является q -квантиль $w = y^{(\ell q)}$, где $y^{(1)}, \dots, y^{(\ell)}$ — вариационный ряд, $q = \frac{C_-}{C_- + C_+}$.

Модель $a(x, w) = \langle x, w \rangle$: задача линейного программирования после замены переменных $\varepsilon_i^+ = (a(x_i) - y_i)_+$, $\varepsilon_i^- = (y_i - a(x_i))_+$:

$$\begin{cases} \sum_{i=1}^{\ell} (C_+ \varepsilon_i^+ + C_- \varepsilon_i^-) \rightarrow \min_w; \\ \langle x_i, w \rangle - y_i = \varepsilon_i^+ - \varepsilon_i^-; \quad \varepsilon_i^+ \geq 0; \quad \varepsilon_i^- \geq 0. \end{cases}$$

Робастная (помехоустойчивая) регрессия

Функция Мешалкина: $L(\varepsilon) = b(1 - \exp(-\frac{1}{b}\varepsilon^2))$, $\varepsilon = a - y$



Модель регрессии: не обязательно линейная $a(x, w)$

Постановка оптимизационной задачи:

$$\sum_{i=1}^{\ell} \exp\left(-\frac{1}{b}(a(x_i, w) - y_i)^2\right) \rightarrow \max_w$$

Численное решение — методом SG или Ньютона-Рафсона

Задачи прогнозирования временных рядов

Дано: $y_0, y_1, \dots, y_t, \dots$ — временной ряд, $y_i \in \mathbb{R}$

Найти: $\hat{y}_{t+d}(w) = f_{t,d}(y_1, \dots, y_t; w)$ — модель временного ряда,
где $d = 1, \dots, D$, D — горизонт прогнозирования,
 w — вектор параметров модели.

Критерий: минимум среднеквадратичной ошибки прогнозов:

$$\sum_{t=T_0}^T (\hat{y}_{t+d}(w) - y_{t+d})^2 \rightarrow \min_w$$

Пример: линейная модель авторегрессии.

В роли признаков выступают n предыдущих наблюдений ряда:

$$\hat{y}_{t+1}(w) = \sum_{j=1}^n w_j y_{t-j+1}, \quad w \in \mathbb{R}^n$$

В роли объектов $\ell = t - n + 1$ моментов истории ряда.

Регуляризаторы, штрафующие сложность линейной модели

Регуляризатор — аддитивная добавка к основному критерию:

$$\sum_{i=1}^{\ell} \mathcal{L}(w, x_i) + \tau \text{штраф}(w) \rightarrow \min_w$$

где τ — коэффициент регуляризации

L₂-регуляризация:

$$\text{штраф}(w) = \|w\|_2^2 = \sum_{j=1}^n w_j^2.$$

L₁-регуляризация (приводит к отбору признаков):

$$\text{штраф}(w) = \|w\|_1 = \sum_{j=1}^n |w_j|.$$

L₀-регуляризация (приводит к отбору признаков):

$$\text{штраф}(w) = \|w\|_0 = \sum_{j=1}^n [w_j \neq 0].$$

Напоминание. L1-регуляризация приводит к отбору признаков

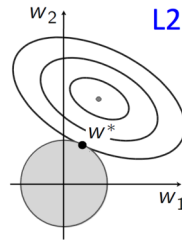
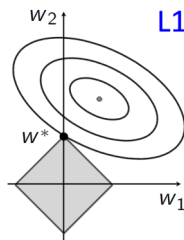
Эквивалентная постановка задачи $\sum_{i=1}^{\ell} \mathcal{L}(w, x_i) \rightarrow \min_w$

с регуляризатором в виде ограничения-неравенства:

$$\text{L1: } \sum_{j=1}^n |w_j| \leq \kappa$$

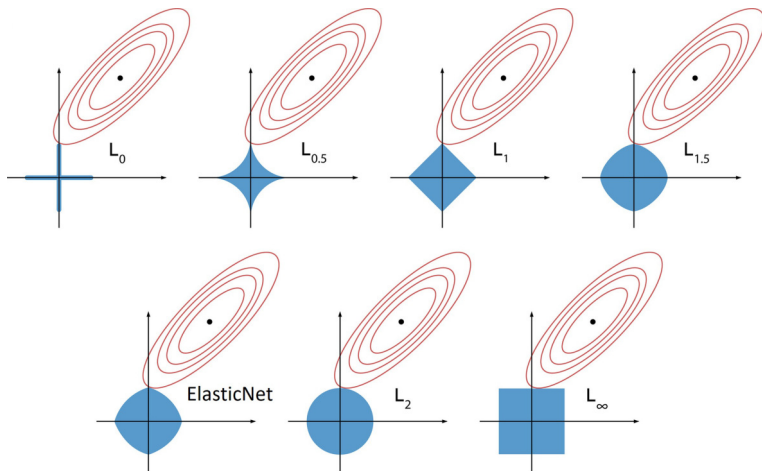
$$\text{L2: } \sum_{j=1}^n w_j^2 \leq \kappa$$

L1 — это метод LASSO (Least Absolute Shrinkage and Selection Operator) приводит к обнулению некоторых w_j , то есть к *отбору признаков*



Напоминание. Геометрический смысл L_p -регуляризаторов

Отбор признаков происходит благодаря негладкости нормы:



Негладкие регуляризаторы для отбора и группировки признаков

Общий вид регуляризаторов (μ — параметр селективности):

$$\sum_{i=1}^{\ell} \mathcal{L}(w, x_i) + \sum_{j=1}^n R_{\mu}(w_j) \rightarrow \min_w.$$

Регуляризаторы с эффектами отбора и группировки признаков:

LASSO (L_1): $R_{\mu}(w) = \mu|w|$

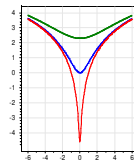
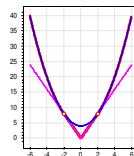
Elastic Net: $R_{\mu}(w) = \mu|w| + \tau w^2$

Support Feature Machine (SFM):

$$R_{\mu}(w) = \begin{cases} 2\mu|w|, & |w| \leq \mu; \\ \mu^2 + w^2, & |w| \geq \mu; \end{cases}$$

Relevance Feature Machine (RFM):

$$R_{\mu}(w) = \ln(\mu w^2 + 1)$$



Задачи ранжирования (Learning to Rank, LtR, L2R, LETOR)

Ранжирование нужно везде, где система предоставляет пользователю выбор из большого числа вариантов:

- выдача поисковой системы
- рекомендации книг, фильмов, музыки, и др. товаров
- рекомендации контента в дистанционном образовании
- автоматическое завершение запроса (auto-suggest)
- варианты ответа в диалоговых системах
- варианты перевода в системах машинного перевода

Критерий конструируется по-разному в трёх подходах:

- Point-wise — поточечный (аналог регрессии/классификации)
- Pair-wise — попарный (качество парных сравнений)
- List-wise — списочный (качество ранжированного списка)

Попарный (pair-wise) подход к обучению ранжированию

Дано: $X^\ell = \{x_1, \dots, x_\ell\}$ — обучающая выборка

$i \prec j$ — правильный порядок на парах объектов (x_i, x_j)
(например, в поиске два документа в ответ на один запрос)

Найти: модель ранжирования $a: X \rightarrow \mathbb{R}$ такую, что

$$i \prec j \Rightarrow a(x_i, w) < a(x_j, w)$$

Критерий: число неверно упорядоченных пар объектов (x_i, x_j)
или аппроксимированный попарный эмпирический риск:

$$\sum_{i \prec j} [a(x_j, w) < a(x_i, w)] \leq \sum_{i \prec j} L(\underbrace{a(x_j, w) - a(x_i, w)}_{M_{ij}(w)}) \rightarrow \min_w$$

где $L(M)$ — убывающая функция *парного отступа* $M_{ij}(w)$

Задача восстановления плотности распределения

Дано: $X^\ell = \{x_1, \dots, x_\ell\}$ — обучающая выборка

Найти: вектор параметров θ в модели $p(x|\theta)$

Критерий: максимум правдоподобия

$$\sum_{i=1}^{\ell} \ln p(x_i|\theta) \rightarrow \max_{\theta}$$

или максимум апостериорной вероятности

$$\sum_{i=1}^{\ell} \ln p(x_i|\theta) + \ln p(\theta|\gamma) \rightarrow \max_{\theta}$$

где γ — вектор гиперпараметров априорного распределения

Задача восстановления смеси плотностей распределения

Дано: $X^\ell = \{x_1, \dots, x_\ell\}$ — обучающая выборка

Найти: параметры w_j, θ_j в модели $p(x|\theta, w) = \sum_{j=1}^K w_j p(x|\theta_j)$

Критерий: максимум правдоподобия

$$\sum_{i=1}^{\ell} \ln p(x_i|\theta, w) \rightarrow \max_{\theta, w}$$

или максимум апостериорной вероятности

$$\sum_{i=1}^{\ell} \ln p(x_i|\theta, w) + \ln p(\theta, w|\gamma) \rightarrow \max_{\theta, w}$$

где γ — вектор гиперпараметров априорного распределения

Задача кластеризации (clustering)

Дано: $X^\ell = \{x_1, \dots, x_\ell\}$ — обучающая выборка, $x_i \in \mathbb{R}^n$

Найти:

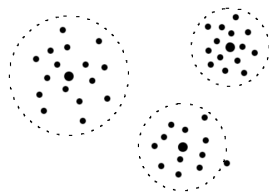
- центры кластеров — параметры $\mu_j \in \mathbb{R}^n$, $j = 1, \dots, K$
- какому кластеру принадлежит каждый объект $a_i \in \{1, \dots, K\}$

Критерий: минимум суммы
внутрикластерных расстояний

$$\sum_{i=1}^{\ell} \|x_i - \mu_{a_i}\|^2 \rightarrow \min_{\{a_i\}, \{\mu_j\}}$$

Метрика, как правило, евклидова
(но может быть и другая):

$$\|x - \mu_j\|^2 = \sum_{d=1}^n (f_d(x) - \mu_{jd})^2$$



Одноклассовая классификация (one-class classification)

Задачи детекции выбросов / аномалий / новизны
(outlier / anomaly / novelty detection)

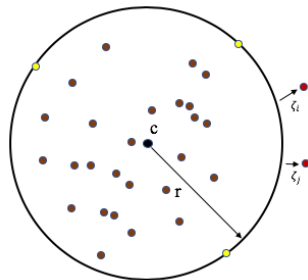
Дано: $X^\ell = \{x_1, \dots, x_\ell\}$ — обучающая выборка, $x_i \in \mathbb{R}^n$

Найти: центр $c \in \mathbb{R}^n$ и радиус r шара, охватывающего всю выборку кроме, быть может, небольшого числа аномальных объектов-выбросов

Критерий: минимизация радиуса шара и суммы штрафов за выход из шара:

$$\tau r^2 + \sum_{i=1}^{\ell} L(\underbrace{r^2 - \|x_i - c\|^2}_{\zeta_i = \text{margin}(c, r)}) \rightarrow \min_{c, r},$$

где $L(M)$ — убывающая функция отступа



Задача частичного обучения (semi-supervised learning, SSL)

Дано:

$X^k = \{x_1, \dots, x_k\}$ — размеченные объекты (labeled data);
 $\{y_1, \dots, y_k\}$, $y_i \in Y$

$U = \{x_{k+1}, \dots, x_\ell\}$ — неразмеченные объекты (unlabeled data).

Найти: классификации $\{a_{k+1}, \dots, a_\ell\}$ неразмеченных объектов

Критерий: без модели классификации (transductive learning):

$$\sum_{i=1}^{\ell} \|x_i - \mu_{a_i}\|^2 + \lambda \sum_{i=1}^k [a_i \neq y_i] \rightarrow \min_{\{a_i\}, \{\mu_j\}}$$

Критерий с моделью классификации $a_i = a(x_i, w)$:

$$\sum_{i=1}^{\ell} \|x_i - \mu_{a_i}\|^2 + \lambda \sum_{i=1}^k \mathcal{L}(w, x_i, y_i) + \tau R(w) \rightarrow \min_{\{a_i\}, \{\mu_j\}, w}$$

где $\mathcal{L}(w, x_i, y_i)$ — функция потерь для модели $a(x_i, w)$

Частный случай SSL: PU-learning (Positive and Unlabeled)

Примеры задач, когда известны объекты только одного класса:

- обнаружение мошеннических транзакций
- рекомендательные системы, персонализация рекламы
- медицинская диагностика при неизвестном анамнезе
- автоматическое пополнение базы знаний фактами

Модель двухклассовой классификации $a(x_i, w) \in \{-1, +1\}$

Неразмеченные трактуются как негативные с весом $\lambda \ll 1$:

$$\sum_{i=1}^k \mathcal{L}(w, x_i, +1) + \lambda \sum_{i=k+1}^{\ell} \mathcal{L}(w, x_i, -1) + \tau R(w) \rightarrow \min_w$$

Неразмеченные могут выбираться случайно (Negative Sampling)

Gang Li. A Survey on Positive and Unlabelled Learning. 2013.

J.Bekker, J.Davis. Learning From Positive and Unlabeled Data: A Survey. 2020.

Задача обучения автокодировщика (AutoEncoder)

Дано: $X^\ell = \{x_1, \dots, x_\ell\}$ — обучающая выборка

Найти модель векторизации, сохраняющую информацию:

$f: X \rightarrow Z$ — кодировщик (encoder), кодовый вектор $z = f(x, \alpha)$

$g: Z \rightarrow X$ — декодировщик (decoder), реконструкция $\hat{x} = g(z, \beta)$

Критерий: точность восстановления объектов $g(f(x_i)) = \hat{x}_i \approx x_i$

$$\sum_{i=1}^{\ell} \mathcal{L}(g(f(x_i, \alpha), \beta), x_i) \rightarrow \min_{\alpha, \beta}$$

Квадратичная функция потерь: $\mathcal{L}(\hat{x}, x) = \|\hat{x} - x\|^2$

Пример. Линейный автокодировщик: $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$

$$f(x, A) = \underset{m \times n}{A} x, \quad g(z, B) = \underset{n \times m}{B} z$$

При $m \ll n$ происходит сжатие данных об объектах

Автокодировщик, частично обучаемый с учителем

Данные: размеченные $(x_i, y_i)_{i=1}^k$, неразмеченные $(x_i)_{i=k+1}^\ell$

Найти: **кодировщик** f , декодировщик g и предиктор \hat{y}
(предсказательную модель классификации, регрессии или др.):

$$\sum_{i=1}^{\ell} \mathcal{L}(g(f(x_i, \alpha), \beta), x_i) + \lambda \sum_{i=1}^k \tilde{\mathcal{L}}(\hat{y}(f(x_i, \alpha), \gamma), y_i) \rightarrow \min_{\alpha, \beta, \gamma}$$

$z_i = f(x_i, \alpha)$ — **кодировщик**

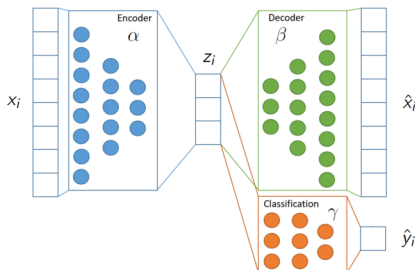
$\hat{x}_i = g(z_i, \beta)$ — **декодировщик**

$\hat{y}_i = \hat{y}(z_i, \gamma)$ — **предиктор**

Функции потерь:

$\mathcal{L}(\hat{x}_i, x_i)$ — реконструкция

$\tilde{\mathcal{L}}(\hat{y}_i, y_i)$ — предсказание



Dor Bank, Noam Koenigstein, Raja Giryes. Autoencoders. 2020

Задачи низкорангового матричного разложения

Дано: матрица $X = \|x_{ij}\|_{\ell \times n}$, $(i, j) \in \Omega \subseteq \{1..\ell\} \times \{1..n\}$

Найти: матрицы $Z = \|z_{it}\|_{\ell \times m}$ и $B = \|b_{tj}\|_{m \times n}$, $m \ll \ell, n$

Критерий: точность восстановления X произведением ZB :

$$\|X - ZB\|_{\Omega} = \sum_{(i,j) \in \Omega} \mathcal{L}\left(x_{ij} - \sum_t z_{it} b_{tj}\right) \rightarrow \min_{Z, B}$$

Применения матричных разложений:

- для восстановления пустых ячеек (missing values) $x_{ij} \notin \Omega$
- для генерации сжатых векторных представлений $x_i \mapsto z_i$
- для векторизации объектов по транзакционным данным
- в рекомендательных системах
- в тематическом моделировании

Графовые (матричные) разложения (graph factorization)

Дано: $(i, j) \in E$ — выборка рёбер графа $\langle V, E \rangle$,

x_{ij} — сходство (близость, similarity) вершин ребра (i, j)

Например, $x_{ij} = [(i, j) \in E]$ — матрица смежности вершин.

Найти: векторные представления вершин $z_i \in \mathbb{R}^d$, чтобы близкие (по графу) вершины имели близкие векторы.

Критерий для неориентированного графа (X симметрична):

$$\|X - ZZ^T\|_E = \sum_{(i,j) \in E} (x_{ij} - \langle z_i, z_j \rangle)^2 \rightarrow \min_Z, \quad Z \in \mathbb{R}^{V \times d}$$

Критерий для ориентированного графа (X несимметрична):

$$\|X - ZB^T\|_E = \sum_{(i,j) \in E} (x_{ij} - \langle z_i, b_j \rangle)^2 \rightarrow \min_{Z, B}, \quad Z, B \in \mathbb{R}^{V \times d}$$

Многомерное шкалирование (multidimensional scaling, MDS)

Дано: $(i, j) \in E$ — выборка рёбер графа $\langle V, E \rangle$,
 x_{ij} — расстояния (distance) между вершинами ребра (i, j)

Найти: векторные представления вершин $z_i \in \mathbb{R}^d$, чтобы
близкие (по графу) вершины имели близкие векторы

Критерий стресса (stress):

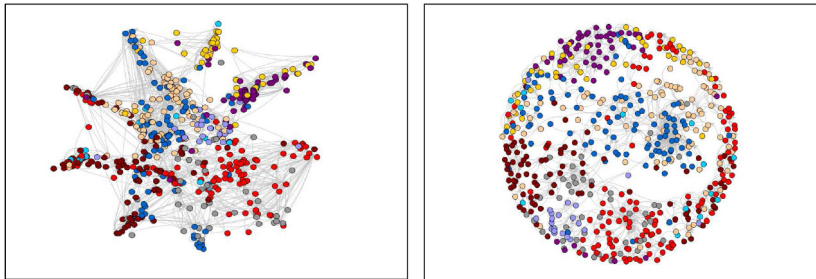
$$\sum_{(i,j) \in E} w_{ij} (\rho(z_i, z_j) - x_{ij})^2 \rightarrow \min, \quad Z \in \mathbb{R}^{V \times d},$$

где $\rho(z_i, z_j) = \|z_i - z_j\|$ — обычно евклидово расстояние,
 w_{ij} — веса (какие расстояния важнее, большие или малые)

Обычно решается методом стохастического градиента (SG)

Многомерное шкалирование для визуализации данных

При $d = 2$ осуществляется проекция выборки на плоскость



- Используется для визуализации кластерных структур
- Форму облака точек можно настраивать весами и метрикой
- Недостаток — искажения неизбежны
- Наиболее популярный метод для визуализации — t-SNE

Laurens van der Maaten, Geoffrey Hinton. Visualizing data using t-SNE. 2008

Перенос обучения (transfer learning)

$z = f(x, \alpha)$ — универсальная часть модели (векторизация)

$y = g(z, \beta)$ — специфичная для задачи часть модели

Базовая задача на выборке $\{x_i\}_{i=1}^{\ell}$ с функцией потерь \mathcal{L}_i :

$$\sum_{i=1}^{\ell} \mathcal{L}_i(g(f(x_i, \alpha), \beta)) \rightarrow \min_{\alpha, \beta}$$

Целевая задача на другой выборке $\{x'_i\}_{i=1}^m$, с другими \mathcal{L}'_i , g' :

$$\sum_{i=1}^m \mathcal{L}'_i(g'(f(x'_i, \alpha), \beta')) \rightarrow \min_{\beta'}$$

при $m \ll \ell$ это может быть намного лучше, чем

$$\sum_{i=1}^m \mathcal{L}'_i(g'(f(x'_i, \alpha), \beta)) \rightarrow \min_{\alpha, \beta}$$

Многозадачное обучение (multi-task learning)

$z = f(x, \alpha)$ — векторизация, универсальная для всех моделей
 $g_t(z, \beta)$ — специфичная часть модели для задачи $t \in T$

Одновременное обучение модели f по задачам X_t , $t \in T$:

$$\sum_{t \in T} \sum_{x_{ti} \in X_t} \mathcal{L}_{ti}(g_t(f(x_{ti}, \alpha), \beta_t)) \rightarrow \min_{\alpha, \{\beta_t\}}$$

Обучаемость (learnability): качество решения отдельной задачи $\langle X_t, \mathcal{L}_t, g_t \rangle$ улучшается с ростом объёма выборки $\ell_t = |X_t|$.

Learning to learn: качество решения каждой из задач $t \in T$ улучшается с ростом как ℓ_t , так и общего числа задач $|T|$.

Few-shot learning: для решения новой задачи t достаточно небольшого числа примеров, иногда даже одного.

M. Crawshaw. Multi-task learning with deep neural networks: a survey. 2020

Y. Wang et al. Generalizing from a few examples: a survey on few-shot learning. 2020

Дистилляция моделей или суррогатное моделирование

Обучение **сложной модели** $a(x, w)$ «долго, дорого»:

$$\sum_{i=1}^{\ell} \mathcal{L}(a(x_i, w), y_i) \rightarrow \min_w$$

Обучение простой модели $b(x, w')$, возможно, на других данных:

$$\sum_{i=1}^k \mathcal{L}(b(x'_i, w'), a(x'_i, w)) \rightarrow \min_{w'}$$

Примеры задач:

- замена сложной модели (климат, аэродинамика и др.), которая вычисляется на суперкомпьютере месяцами, «лёгкой» аппроксимирующей суррогатной моделью
- замена сложной нейросети, которая обучается неделями на больших данных, «лёгкой» аппроксимирующей нейросетью с минимизацией числа нейронов и связей

Задача обучения с привилегированной информацией

x_i^* — информация об объекте x_i , доступная только на обучении

Раздельное обучение модели-ученика и **модели-учителя**:

$$\sum_{i=1}^{\ell} \mathcal{L}(a(x_i, w), y_i) \rightarrow \min_w \quad \sum_{i=1}^{\ell} \mathcal{L}(a(x_i^*, w^*), y_i) \rightarrow \min_{w^*}$$

Модель-ученик обучается у **модели-учителя**:

$$\sum_{i=1}^{\ell} \mathcal{L}(a(x_i, w), y_i) + \mu \mathcal{L}(a(x_i, w), a(x_i^*, w^*)) \rightarrow \min_w$$

Совместное обучение модели-ученика и **модели-учителя**:

$$\begin{aligned} \sum_{i=1}^{\ell} \mathcal{L}(a(x_i, w), y_i) + \lambda \mathcal{L}(a(x_i^*, w^*), y_i) + \\ + \mu \mathcal{L}(a(x_i, w), a(x_i^*, w^*)) \rightarrow \min_{w, w^*} \end{aligned}$$

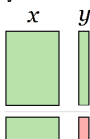
D.Lopez-Paz, L.Bottou, B.Scholkopf, V.Vapnik. Unifying distillation and privileged information. 2016.

Обучение с использованием привилегированной информации

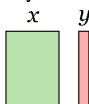
x_i^* — информация об объекте x_i , доступная только на обучении

Варианты LUPI (Learning Using Privileged Information):

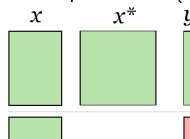
с учителем



без учителя



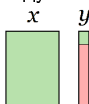
привилегированное (LUPI)



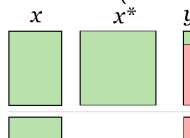
частичное



трандуктивное



частичное (SLUPI)



V. Vapnik, A. Vashist. A new learning paradigm: Learning Using Privileged Information // Neural Networks. 2009.

- ❶ Предварительная обработка (data preparation)
 - извлечение признаков (feature extraction)
 - отбор признаков (feature selection)
 - восстановление пропусков (missing values)
 - фильтрация выбросов (outlier detection)
- ❷ Обучение с учителем (supervised learning)
 - классификация (classification)
 - регрессия (regression)
 - ранжирование (learning to rank)
 - прогнозирование (forecasting)
- ❸ Обучение без учителя (unsupervised learning)
 - восстановление плотности (density estimation)
 - кластеризация (clustering)
 - одноклассовая классификация (OCC, anomaly detection)
 - поиск ассоциативных правил (association rule learning)
- ❹ Частичное обучение (semi-supervised learning)
 - трансдуктивное обучение (transductive learning)
 - обучение с положительными примерами (PU-learning)

- 5 Обучаемая векторизация объектов (representation learning)
 - автокодировщики (autoencoders, feature learning)
 - матричные разложения (matrix factorization)
 - обучение многообразий (manifold learning)
- 6 Перенос обучения (transfer learning)
- 7 Многозадачное обучение (multitask learning)
- 8 Привилегированное обучение (privileged learning, distilling)
- 9 Инкрементное обучение (online/incremental learning)
- 10 Активное обучение (active learning)
- 11 Обучение с подкреплением (reinforcement learning)
- 12 Мета-обучение (meta-learning, AutoML)
- 13 Обучение близости/связей (similarity/relational learning)
- 14 Обучение структуры модели (structure learning)
- 15 Глубокое обучение (deep learning)
 - Порождение структурированных данных (structured output)
 - Состязательное обучение (adversarial learning)