Обзорная лекция. Постановки задач оптимизации в машинном обучении

Воронцов Константин Вячеславович
vokov@forecsys.ru
http://www.MachineLearning.ru/wiki?title=User:Vokov

Этот курс доступен на странице вики-ресурса
http://www.MachineLearning.ru/wiki
«Машинное обучение (курс лекций, К.В.Воронцов)»

МФТИ • 30 апреля 2021

Содержание

Общая оптимизационная задача машинного обучения

Дано: выборка объектов $\{x_i\}_{i=1}^{\ell}$

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

Критерий: минимум эмпирического риска

$$\sum_{i=1}^{\ell} L_i(w) \;\to\; \min_w$$

где $L_i(w)$ — функция потерь модели $a(x, w)$ на объекте x_i,

или минимум регуляризованного эмпирического риска

$$\sum_{i=1}^{\ell} L_i(w) + \sum_{j=1}^{r} \tau_j R_j(w) \;\to\; \min_w$$

где R_j — регуляризаторы, τ_j — коэффициенты регуляризации

Оптимизационная задача восстановления регрессии

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

1. Фиксируется модель регрессии, например, *линейная*:
$$a(x, w) = \langle x, w \rangle = \sum_{j=1}^n w_j f_j(x), \qquad w \in \mathbb{R}^n$$
2. Фиксируется функция потерь, например, *квадратичная*:
$$L_i(w) = \big(a(x_i, w) - y_i\big)^2$$
3. Метод обучения — *метод наименьших квадратов*:
$$\sum_{i=1}^\ell \big(a(x_i, w) - y_i\big)^2 \to \min_w$$
4. Проверка по тестовой выборке $X^k = (\tilde{x}_i, \tilde{y}_i)_{i=1}^k$:
$$Q(X^k) = \frac{1}{k} \sum_{i=1}^k \big(a(\tilde{x}_i, w) - \tilde{y}_i\big)^2$$

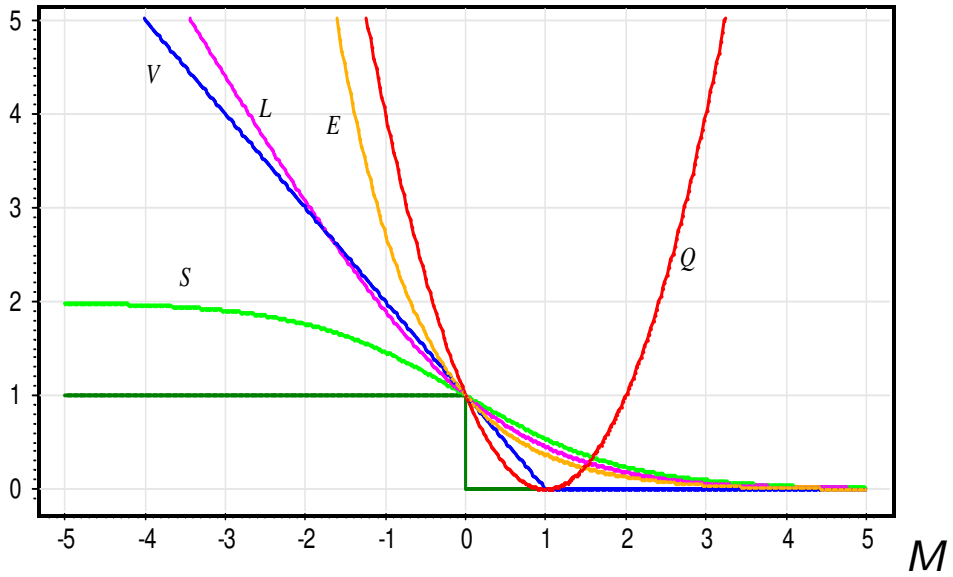
Оптимизационная задача обучения классификация

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

1. Фиксируется модель классификации, например, *линейная*:
$$a(x, w) = \text{sign}\langle x, w\rangle = \text{sign} \sum_{j=1}^n w_j f_j(x)$$
2. Функция потерь — пороговая или её верхняя оценка:
$$L_i(w) = \big[a(x_i, w)y_i < 0\big] = \big[\langle x_i, w\rangle y_i < 0\big] \leqslant \mathscr{L}\big(\langle x_i, w\rangle y_i\big)$$
3. Метод обучения — *минимизация эмпирического риска*:
$$\sum_{i=1}^\ell \big[\langle x_i, w\rangle y_i < 0\big] \leqslant \sum_{i=1}^\ell \mathscr{L}\big(\langle x_i, w\rangle y_i\big) \to \min_w$$
4. Проверка по тестовой выборке $X^k = (\tilde{x}_i, \tilde{y}_i)_{i=1}^k$:
$$Q(X^k) = \frac{1}{k} \sum_{i=1}^k \big[\langle \tilde{x}_i, w\rangle \tilde{y}_i < 0\big]$$

Непрерывные верхние оценки пороговой функции потерь

Часто используемые непрерывные функции потерь $\mathscr{L}(M)$:



$[M < 0]$ — пороговая функция потерь

$V(M) = (1 - M)_+$ — кусочно-линейная (SVM)

$H(M) = (-M)_+$ — кусочно-линейная (Hebb's rule)

$L(M) = \log_2(1 + e^{-M})$ — логарифмическая (LR)

$Q(M) = (1 - M)^2$ — квадратичная (FLD)

$S(M) = 2(1 + e^M)^{-1}$ — сигмоидная (ANN)

$E(M) = e^{-M}$ — экспоненциальная (AdaBoost)

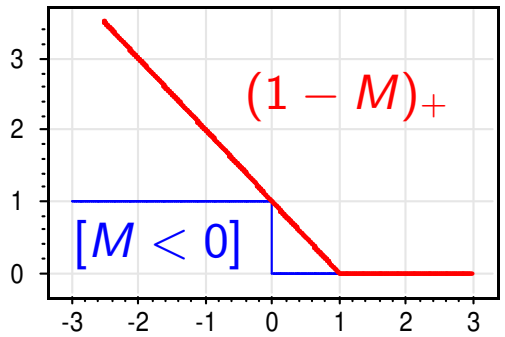
Метод опорных векторов SVM (двухклассовый)

$M_i(w, w_0) = y_i\big(\langle w, x_i\rangle - w_0\big)$ — отступ в линейной модели

Кусочно-линейная функция потерь:

$$\sum_{i=1}^{\ell}\big(1 - M_i(w, w_0)\big)_+ + \frac{1}{2C}\|w\|^2 \;\to\; \min_{w, w_0}$$

- *Функция потерь* штрафует объекты за приближение к границе классов
- *Регуляризация* максимизирует зазор между классами и штрафует за мультиколлинеарность



Важнейшие свойства SVM:

- Задача выпуклого программирования, решение единственно
- Решение *разрежено* — зависит только от *опорных объектов*
- Обобщение на нелинейные модели: $\langle x, x_i\rangle \to K(x, x_i)$

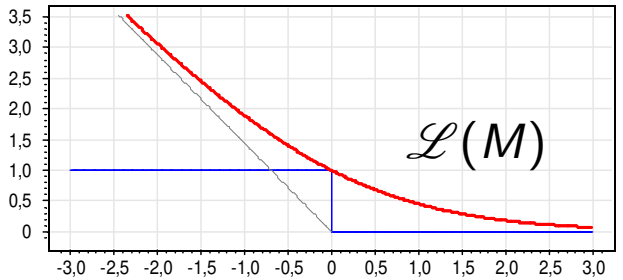
Логистическая регрессия (двухклассовая)

Линейная модель классификации $a(x, w) = \text{sign}\langle x, w\rangle$
Логарифмическая функция потерь:

$$\sum_{i=1}^{\ell} \ln\left(1 + \exp(-\langle w, x_i\rangle y_i)\right) + \frac{\tau}{2}\|w\|^2 \;\to\; \min_w$$

Логарифмическая функция потерь:

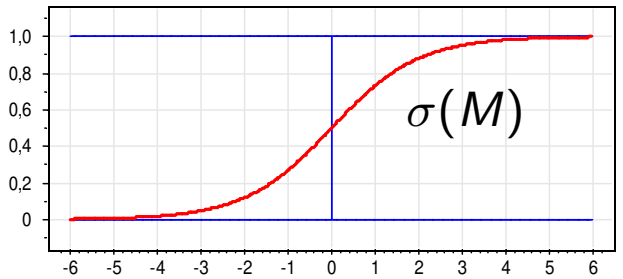
$$\mathscr{L}(M) = \ln\left(1 + e^{-M}\right)$$



Модель условной вероятности:

$$P(y|x, w) = \sigma(M) = \frac{1}{1+e^{-M}},$$

где $\sigma(M)$ — сигмоидная функция



Логистическая регрессия (многоклассовая)

Линейный классификатор при произвольном числе классов $|Y|$:

$$a(x, w) = \arg\max_{y \in Y} \langle w_y, x \rangle, \quad x, w_y \in \mathbb{R}^n$$

Вероятность того, что объект x относится к классу y:

$$P(y|x, w) = \frac{\exp\langle w_y, x \rangle}{\sum_{z \in Y} \exp\langle w_z, x \rangle} = \underset{y \in Y}{\text{SoftMax}} \langle w_y, x \rangle,$$

где SoftMax: $\mathbb{R}^Y \to \mathbb{R}^Y$ переводит произвольный вектор в нормированный вектор дискретного распределения.

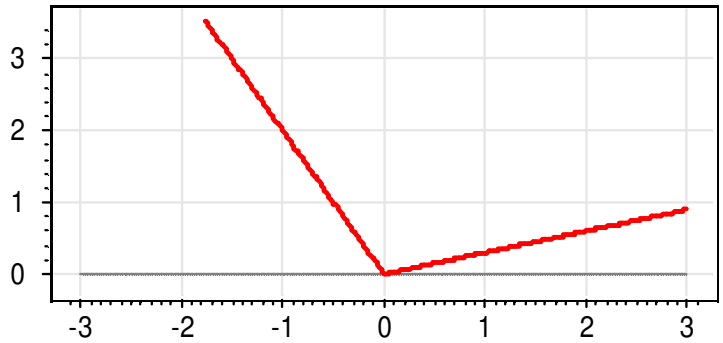
Максимизация правдоподобия (log-loss) с регуляризацией:

$$-\sum_{i=1}^{\ell} \ln P(y_i|x_i, w) + \frac{\tau}{2} \sum_{y \in Y} \|w_y\|^2 \to \min_w.$$

Квантильная регрессия

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

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



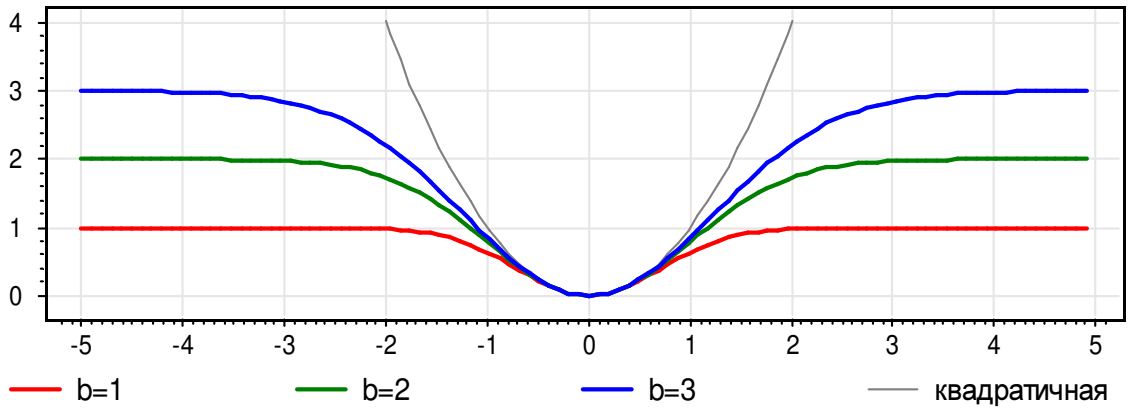
Модель регрессии: линейная $a(x_i, w) = \langle x_i, w \rangle$.

Сведение к задаче линейного программирования:
замена переменных $\varepsilon_i^+ = \big(a(x_i) - y_i\big)_+$, $\varepsilon_i^- = \big(y_i - a(x_i)\big)_+$;

$$\begin{cases} \sum\limits_{i=1}^{\ell} C_+\varepsilon_i^+ + C_-\varepsilon_i^- \to \min\limits_w; \\ \langle x_i, w \rangle - y_i = \varepsilon_i^+ - \varepsilon_i^-; \\ \varepsilon_i^+ \geqslant 0; \quad \varepsilon_i^- \geqslant 0. \end{cases}$$

Робастная регрессия

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



Модель регрессии: $a(x, w)$

Постановка оптимизационной задачи:

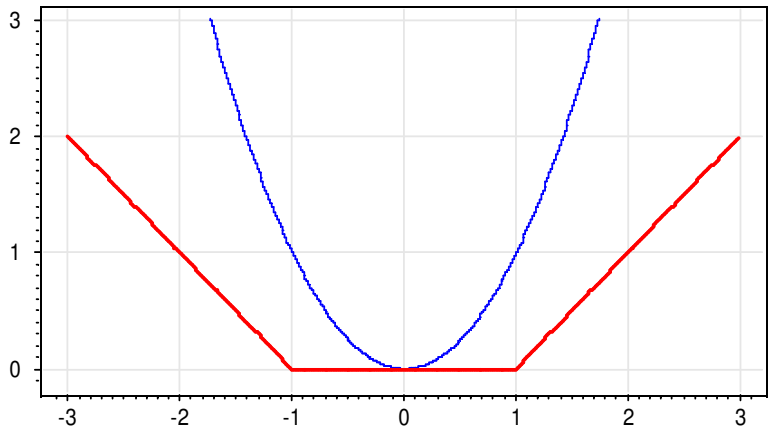
$$\sum_{i=1}^{\ell} \exp\Big(-\frac{1}{b}\big(a(x_i, w) - y_i\big)^2\Big) \to \max_w$$

Численное решение — методом Ньютона-Рафсона

SVM-регрессия

Модель регрессии: $a(x) = \langle x, w \rangle - w_0, \quad w \in \mathbb{R}^n, \quad w_0 \in \mathbb{R}.$

Функция потерь: $\mathscr{L}(\varepsilon) = \big(|\varepsilon| - \delta\big)_+$



Постановка оптимизационной задачи:

$$\sum_{i=1}^{\ell} \big(|\langle w, x_i \rangle - w_0 - y_i| - \delta\big)_+ + \frac{1}{2C}\|w\|^2 \to \min_{w, w_0}.$$

Сводится к выпуклой задаче квадратичного программирования

Регуляризаторы, штрафующие сложность линейной модели

Регуляризатор — аддитивная добавка к основному критерию:

$$\sum_{i=1}^{\ell} \mathscr{L}\big(\langle x_i, w\rangle, y_i\big) + \tau\, \text{штраф}(w) \;\to\; \min_w$$

где $\mathscr{L}(a, y)$ — функция потерь, τ — коэффициент регуляризации

L_2-регуляризация (гребневая регрессия, SVM):

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

L_1-регуляризация (LASSO, ElasticNet — для отбора признаков):

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

L_0-регуляризация (критерии Акаике AIC, байесовский BIC):

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

Негладкие регуляризаторы для отбора признаков

Общий вид регуляризаторов (μ — параметр селективности):

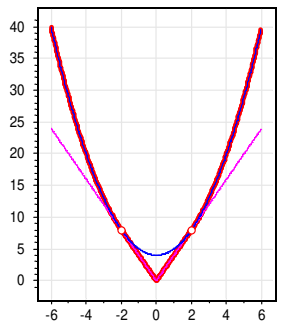
$$\sum_{i=1}^{\ell} \mathscr{L}\big(\langle x_i, w\rangle, y_i\big) + \tau \sum_{j=1}^{n} R_\mu(w_j) \;\to\; \min_w .$$

Регуляризаторы с эффектом группировки зависимых признаков:

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

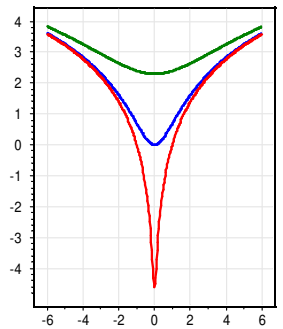
Support Features Machine (SFM):

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



Relevance Features Machine (RFM):

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



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)

Поточечный подход: ранговая регрессия (Ordinal Regression)

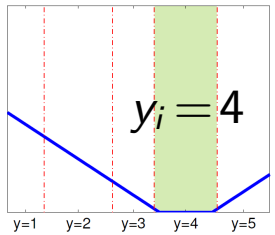
Обучающая выборка $(x_i, y_i)_{i=1}^{\ell}$, где $y_i \in Y = \{1 \prec 2 \prec \cdots \prec K\}$.
Функция ранжирования с параметрами w
и порогами $b_0 = -\infty$, $b_1, \ldots, b_{K-1}$, $b_K = +\infty$:

$$a(x, w, b) = y, \quad \text{если} \quad b_{y-1} < g(x, w) \leqslant b_y$$

Функция потерь $\mathscr{L}(M)$ — убывающая функция отступа M

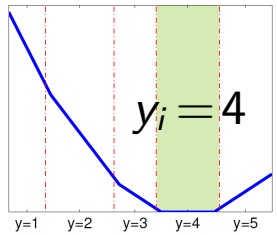
Критерий обучения по двум ближайшим порогам:

$$\sum_{i=1}^{\ell} \mathscr{L}\big(g(x_i, w) - b_{y_i-1}\big) + \mathscr{L}\big(b_{y_i} - g(x_i, w)\big) \to \min_{w,b}$$



Критерий обучения по всем порогам:

$$\sum_{i=1}^{\ell} \sum_{y=1}^{K} \mathscr{L}\Big(\big(b_y - g(x_i, w)\big)\,\mathrm{sign}\big(y - y_i\big)\Big) \to \min_{w,b}$$



J.D.M.Rennie, N.Srebro. Loss functions for preference levels: regression with discrete ordered labels. IJCAI-2005.

Попарный (pair-wise) подход к обучению ранжированию

Дано: $X^\ell = \{x_1, \ldots, x_\ell\}$ — обучающая выборка
$i \prec j$ — правильный порядок на парах (x_i, x_j)

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

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

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

$$\sum_{i \prec j} \big[a(x_j, w) < a(x_i, w)\big] \;\leqslant\; \sum_{i \prec j} \mathscr{L}\big(\underbrace{a(x_j, w) - a(x_i, w)}_{M_{ij}(w)}\big) \;\to\; \min_w$$

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

Списочный (list-wise) подход на основе попарного

Метод стохастического градиента для попарного критерия:

$$w := w - \eta \mathscr{L}'(M_{ij}(w))\, M'_{ij}(w)$$

Q — негладкий критерий, вычисляемый по списку объектов x_i, ранжированному в порядке убывания значений $a(x_i)$.

Примеры негладких критериев Q: MAP, NDCG, pFound и др.

ΔQ_{ij} — изменение Q при перестановке $x_i \leftrightarrows x_j$ в списке.

LambdaRank: домножение градиента на $|\Delta Q_{ij}|$ приводит к приближённой оптимизации негладкого критерия Q:

$$w := w - \eta \mathscr{L}'(M_{ij}(w))\, M'_{ij}(w) \cdot |\Delta Q_{ij}|$$

C.Burges. From RankNet to LambdaRank to LambdaMART: an overview. 2010

Задача восстановления плотности распределения

Дано: обучающая выборка $\{x_i : i = 1, \ldots, \ell\}$

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

Критерий: максимум правдоподобия

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

или максимум апостериорной вероятности

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

где γ — вектор гиперпараметров априорного распределения

Задача восстановления смеси плотностей распределения

Дано: обучающая выборка $\{x_i : i = 1, \ldots, \ell\}$

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

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

или максимум апостериорной вероятности

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

где γ — вектор гиперпараметров априорного распределения

Задача кластеризации (clustering)

Дано: обучающая выборка $\{x_i \in \mathbb{R}^n : i = 1, \ldots, \ell\}$

Найти:

— центры кластеров $\mu_j \in \mathbb{R}^n$, $j = 1, \ldots, K$

— какому кластеру принадлежит каждый объект $a_i \in \{1, \ldots, K\}$

Критерий: минимум внутрикластерных расстояний

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

в случае евклидовой метрики

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

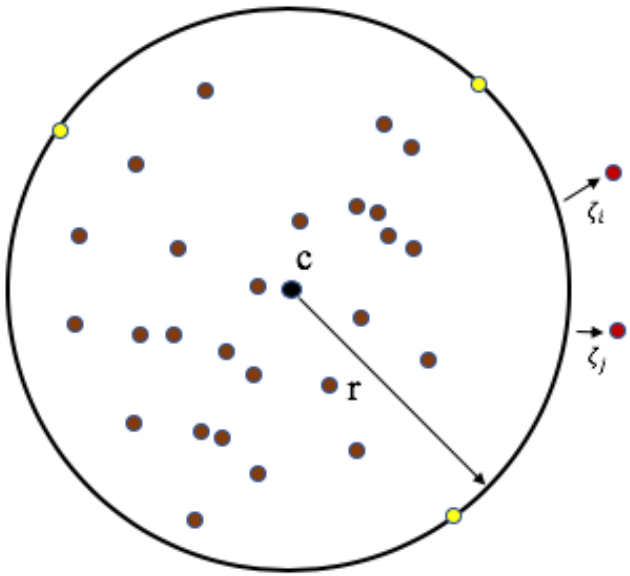
Одноклассовый SVM (one-class SVM, OSVM)

Дано: обучающая выборка $\{x_i \in \mathbb{R}^n : i = 1, \ldots, \ell\}$

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

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

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



При $\mathscr{L}(\zeta) = (-\zeta)_+$ свойства решения аналогичны SVM:

- Выпуклая задача квадратичного программирования
- Решение *разрежено* — зависит только от *опорных объектов*
- Обобщение на нелинейные модели: $\langle x_i, x_j \rangle \to K(x_i, x_j)$

Задача частичного обучения (semi-supervised learning, SSL)

Дано:

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

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

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

Критерий без модели классификации (transductive learning):

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

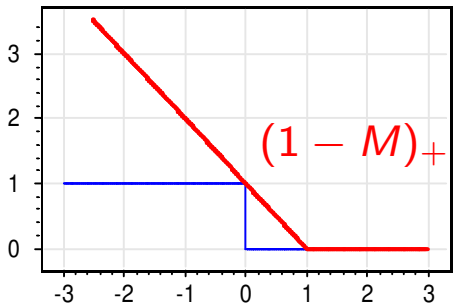
При построении модели классификации, $a_i = a(x_i, w)$:

$$\sum_{i=1}^{\ell} \|x_i - \mu_{a_i}\|^2 + \lambda \sum_{i=1}^{k} \mathscr{L}(a(x_i, w), y_i) \;\to\; \min_{\{a_i\},\{\mu_j\},w}$$

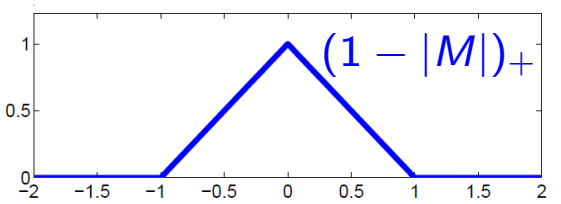
Метод TSVM — трансдуктивный SVM

$M_i = (\langle w, x_i \rangle - w_0) y_i$ — отступ объекта x_i

- Функция потерь $\mathscr{L}(M) = (1 - M)_+$ штрафует за уменьшение отступа



- Функция потерь $\mathscr{L}(M) = (1 - |M|)_+$ штрафует за попадание объекта внутрь разделяющей полосы



Обучение весов w, w_0 по частично размеченной выборке:

$$\sum_{i=1}^{k} (1 - M_i(w, w_0))_+ + \gamma \sum_{i=1}^{\ell} (1 - |M_i(w, w_0)|)_+ + \frac{\|w\|^2}{2C} \to \min_{w, w_0}$$

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 $a(x_i, w)$.
Неразмеченные трактуются как негативные с весом $C_- \ll C_+$:

$$C_+ \sum_{i=1}^{k} \mathscr{L}(a(x_i, w), +1) + C_- \sum_{i=k+1}^{\ell} \mathscr{L}(a(x_i, w), -1) + R(w) \to \min_w$$

Один из успешных методов — Biased SVM.

Gang Li. A Survey on Positive and Unlabelled Learning. 2013.
J.Bekker, J.Davis. Learning From Positive and Unlabeled Data: A Survey. 2020.

Задачи низкорангового матричного разложения

- Понижение размерности для классификации/регрессии
- Формирование векторных представлений объектов
- Восстановление пропущенных значений в матрице

Дано: матрица $Z = \|z_{ij}\|_{n\times m}$, $(i,j) \in \Omega \subseteq \{1..n\} \times \{1..m\}$

Найти: матрицы $X = \|x_{it}\|_{n\times k}$ и $Y = \|y_{tj}\|_{k\times m}$ такие, что

$$\|Z - XY\| = \sum_{(i,j)\in\Omega} \mathscr{L}\left(z_{ij} - \sum_t x_{it}y_{tj}\right) \to \min_{X,Y}$$

Почему на практике отказываются от классического SVD:

- неквадратичная функция потерь $\mathscr{L}$
- неотрицательное матричное разложение: $x_{it} \geqslant 0$, $y_{tj} \geqslant 0$
- разреженные данные: $|\Omega| \ll nm$
- ортогональность не нужна или не интерпретируема

Задача построения автокодировщика (обучение без учителя)

$X^\ell = \{x_1, \ldots, x_\ell\}$ — обучающая выборка
$f: X \to Z$ — кодировщик (encoder), кодовый вектор $z = f(x, \alpha)$
$g: Z \to X$ — декодировщик (decoder), реконструкция $\hat{x} = g(z, \beta)$

Суперпозиция $\hat{x} = g(f(x))$ должна восстанавливать исходные x_i:

$$\mathscr{L}_{\text{AE}}(\alpha, \beta) = \sum_{i=1}^{\ell} \mathscr{L}\big(g(f(x_i, \alpha), \beta), x_i\big) \to \min_{\alpha,\beta}$$

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

Пример 1. Линейный автокодировщик: $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$

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

Пример 2. Двухслойная сеть с функциями активации σ_f, σ_g:

$$f(x, A) = \sigma_f(Ax + a), \qquad g(z, B) = \sigma_g(Bz + b)$$

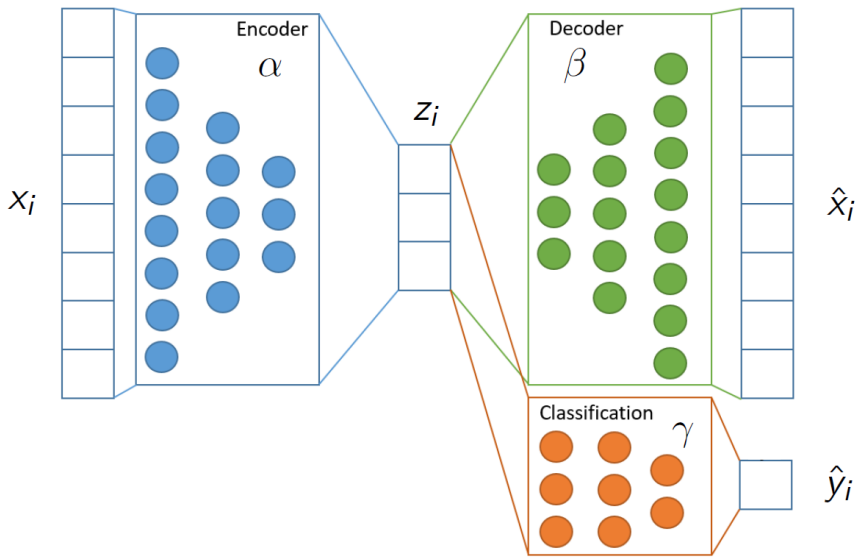
Автокодировщики для обучения с учителем

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

Совместное обучение кодировщика f, декодировщика g и предсказательной модели (классификации, регрессии или др.):

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

$z_i = f(x_i,\alpha)$ — кодировщик
$\hat{x}_i = g(z_i,\beta)$ — декодировщик
$\hat{y}_i = \hat{y}(z_i,\gamma)$ — предиктор



Функции потерь:
$\mathscr{L}(\hat{x}_i, x_i)$ — реконструкция
$\tilde{\mathscr{L}}(\hat{y}_i, y_i)$ — предсказание

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

Многомерное шкалирование (multidimensional scaling, MDS)

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

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

Критерий стресса (stress):

$$\sum_{(i,j)\in E} w(R_{ij})\big(\rho(z_i,z_j) - R_{ij}\big)^2 \to \min_Z, \quad Z \in \mathbb{R}^{V\times d},$$

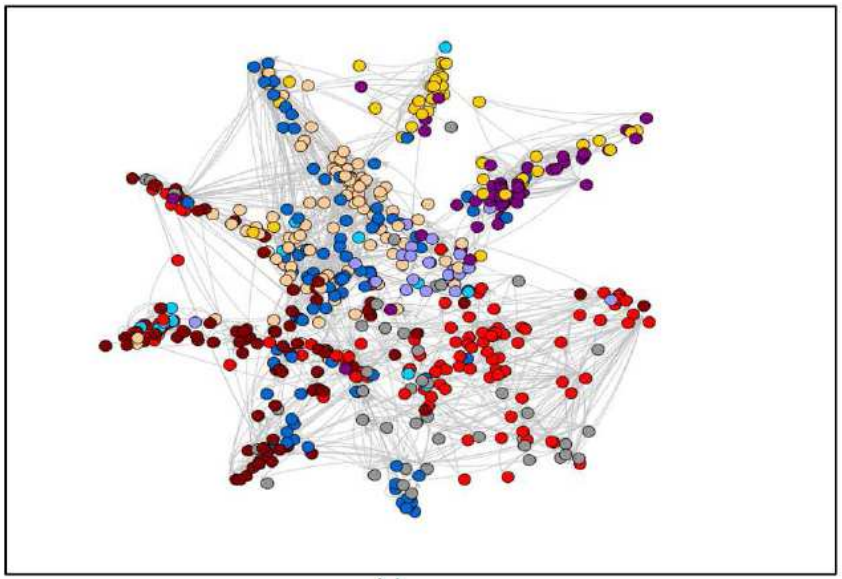
где $\rho(z_i,z_j) = \|z_i - z_j\|$ — обычно евклидово расстояние,
$w(R_{ij})$ — веса (какие расстояния важнее, большие или малые).

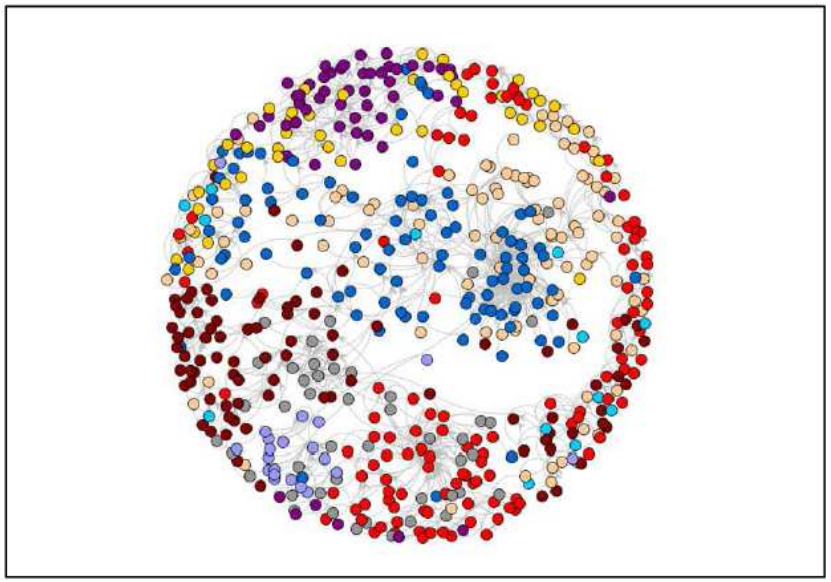
Обычно решается методом стохастического градиента (SG).

I.Chami et al. Machine learning on graphs: a model and comprehensive taxonomy. 2020.

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

Графовые (матричные) разложения (graph factorization)

Дано: $(i,j) \in E$ — выборка рёбер графа $\langle V, E \rangle$,
S_{ij} — близость между вершинами ребра (i,j).
Например, $S_{ij} = [(i,j) \in E]$ — матрица смежности вершин.

Найти: векторные представления вершин, так, чтобы близкие (по графу) вершины имели близкие векторы.

Критерий для неориентированного графа (S симметрична):

$$\sum_{(i,j)\in E} \big(\langle z_i, z_j \rangle - S_{ij}\big)^2 \to \min_Z, \quad Z \in \mathbb{R}^{V\times d}$$

Критерий для ориентированного графа (S несимметрична):

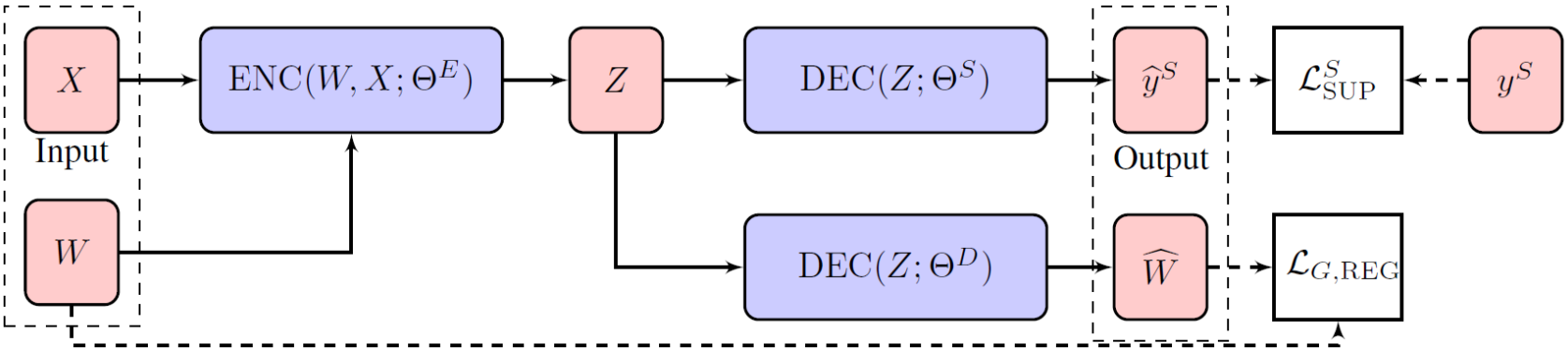
$$\sum_{(i,j)\in E} \big(\langle \varphi_i, \theta_j \rangle - S_{ij}\big)^2 \to \min_{\Phi,\Theta}, \quad \Phi, \Theta \in \mathbb{R}^{V\times d}$$

Обычно решается методом стохастического градиента (SG).

I.Chami et al. Machine learning on graphs: a model and comprehensive taxonomy. 2020.

GraphEDM: обобщённый автокодировщик на графах

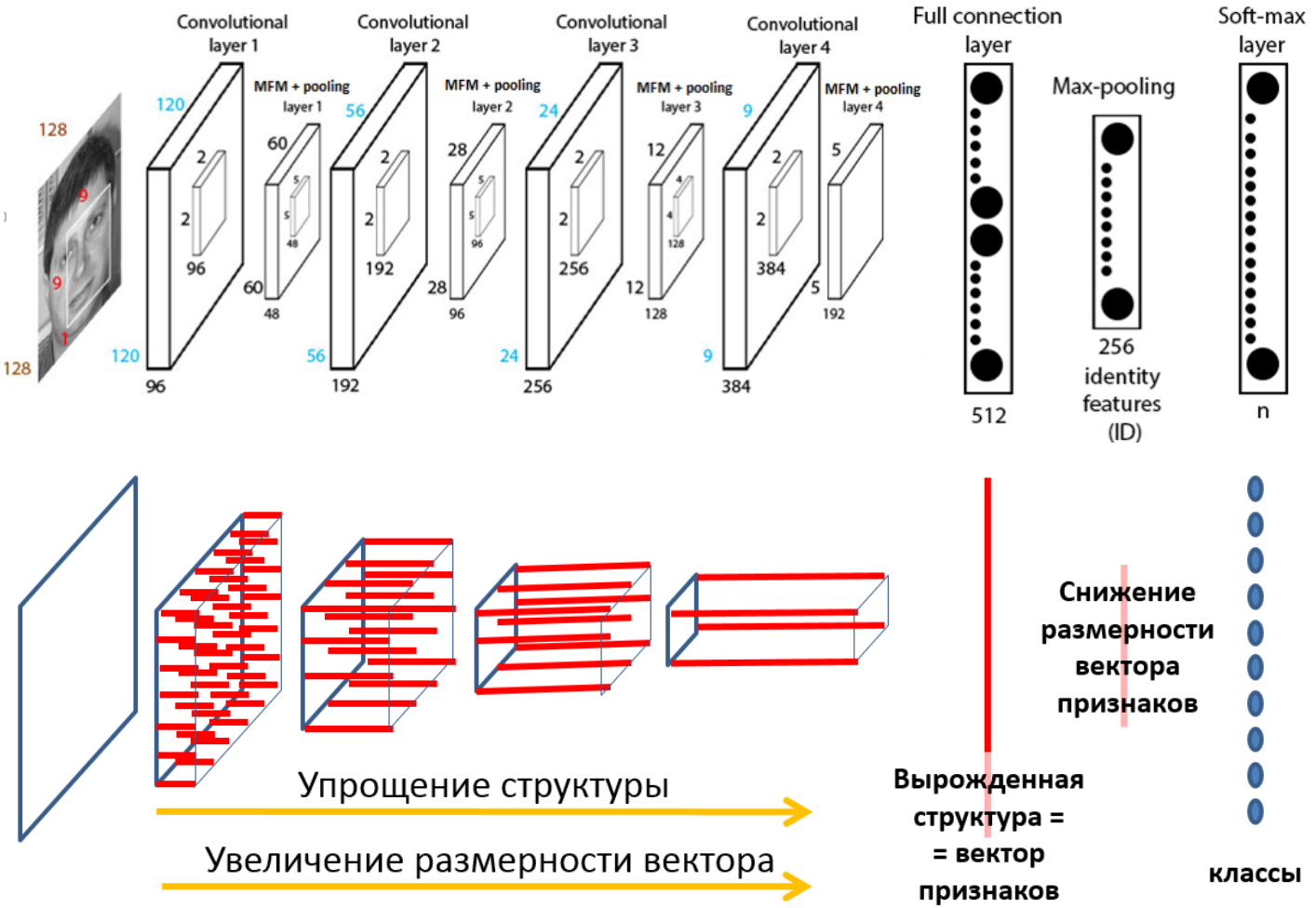
Graph Encoder Decoder Model — обобщает более 30 моделей:



$W \in \mathbb{R}^{V \times V}$ — входные данные о рёбрах
$X \in \mathbb{R}^{V \times n}$ — входные данные о вершинах, признаковые описания
$Z \in \mathbb{R}^{V \times d}$ — векторные представления вершин графа
$\mathrm{DEC}(Z; \Theta^D)$ — декодер, реконструирующий данные о рёбрах
$\mathrm{DEC}(Z; \Theta^S)$ — декодер, решающий supervised-задачу
y^S — (semi-)supervised данные о вершинах или рёбрах
$\mathcal{L}$ — функции потерь

I.Chami et al. Machine learning on graphs: a model and comprehensive taxonomy. 2020.

Глубокая свёрточная сеть как способ векторизации изображений

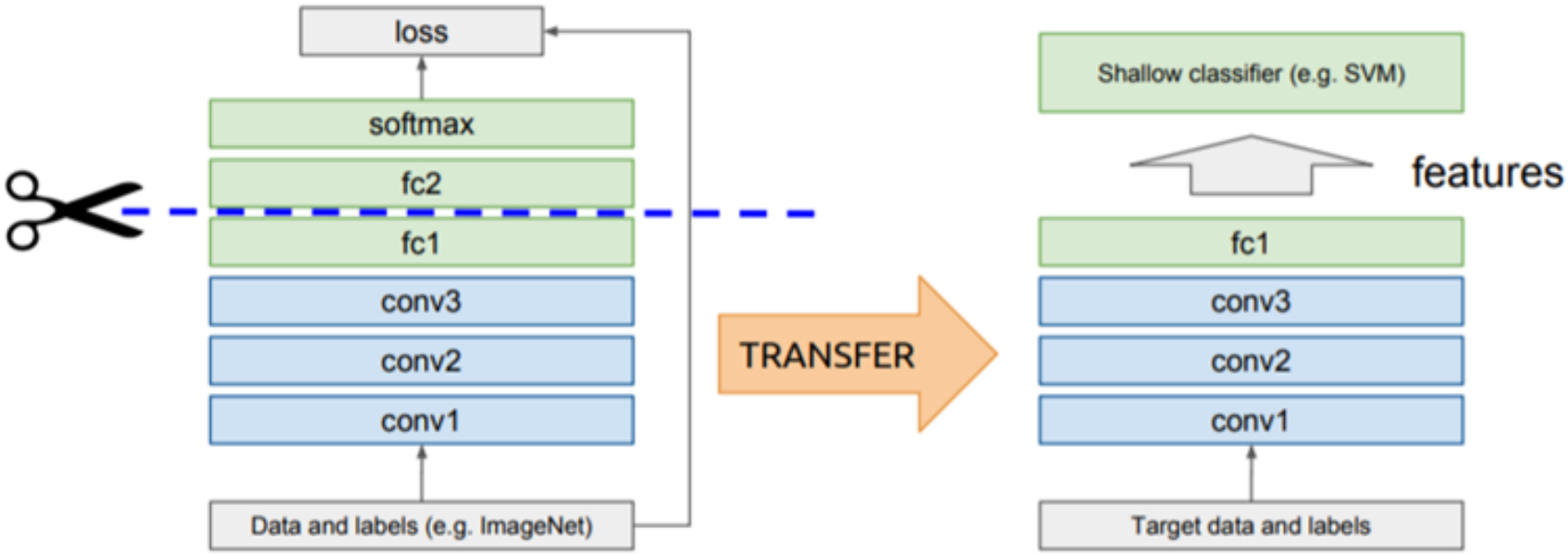


Визильтер Ю.В., Горбацевич В.С. Структурно-функциональный анализ и синтез глубоких конволюционных нейронных сетей. ММРО-2017.

Пред-обучение нейронных сетей (pre-training)

Свёрточная сеть для обработки изображений:

- $z = f(x, \alpha)$ — свёрточные слои для векторизации объектов
- $y = g(z, \beta)$ — полносвязные слои под конкретную задачу



Jason Yosinski, Jeff Clune, Yoshua Bengio, Hod Lipson. How transferable are features in deep neural networks? 2014.

Перенос обучения (transfer learning)

$f(x, \alpha)$ — универсальная часть модели (векторизация)
$g(x, \beta)$ — специфичная для задачи часть модели

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

$$\sum_{i=1}^{\ell} \mathscr{L}_i\big(f(x_i, \alpha), g(x_i, \beta)\big) \;\to\; \min_{\alpha,\beta}$$

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

$$\sum_{i=1}^{m} \mathscr{L}'_i\big(f(x'_i, \alpha), g'(x'_i, \beta')\big) \;\to\; \min_{\beta'}$$

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

$$\sum_{i=1}^{m} \mathscr{L}'_i\big(f(x'_i, \alpha), g'(x'_i, \beta')\big) \;\to\; \min_{\alpha,\beta'}$$

Sinno Jialin Pan, Qiang Yang. A Survey on Transfer Learning. 2009

Многозадачное обучение (multi-task learning)

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

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

$$\sum_{t \in T} \sum_{i \in X_t} \mathscr{L}_{ti}\big(f(x_{ti}, \alpha), g_t(x_{ti}, \beta_t)\big) \;\to\; \min_{\alpha, \{\beta_t\}}$$

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

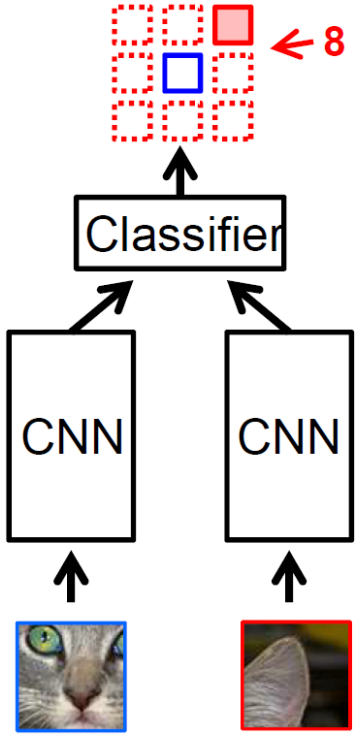
Learning to learn: качество решения каждой из задач $t \in T$ улучшается с ростом как ℓ_t, так и общего числа задач $|T|$.

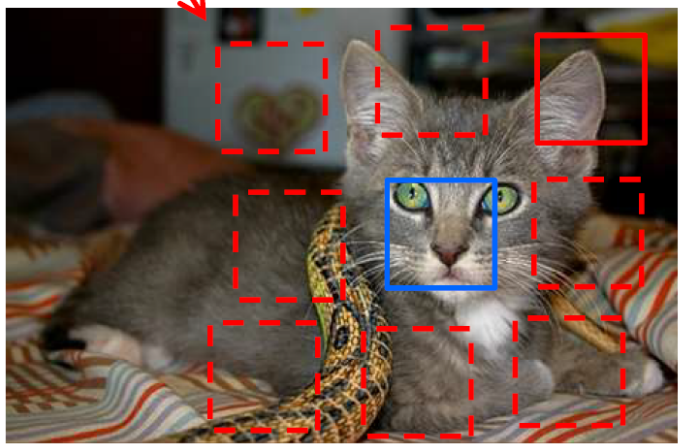
Few-shot learning: для решения задачи t достаточно небольшого числа примеров, иногда даже одного.

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

Самостоятельное обучение (self-supervised learning)

Модель векторизации $z = f(x, \alpha)$ обучается предсказывать взаимное расположение пар фрагментов одного изображения





Unsupervised visual representation learning by context prediction,
Carl Doersch, Abhinav Gupta, Alexei A. Efros, ICCV 2015

Преимущество: сеть выучивает векторные представления объектов без размеченной обучающей выборки (без ImageNet).

Дистилляция моделей или суррогатное моделирование

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

$$\sum_{i=1}^{\ell} \mathscr{L}\big(a(x_i, w), y_i\big) \;\to\; \min_{w}$$

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

$$\sum_{i=1}^{k} \mathscr{L}\big(b(x'_i, w'), a(x'_i, w)\big) \;\to\; \min_{w'}$$

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:

$$\sum_{i=1}^{\ell} \mathscr{L}\big(a(x_i, w), y_i\big) \to \min_w \qquad \sum_{i=1}^{\ell} \mathscr{L}\big(a(x_i^*, w^*), y_i\big) \to \min_w$$

Модель-ученик обучается повторять ошибки модели-учителя:

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

Совместное обучение модели-ученика и модели-учителя:

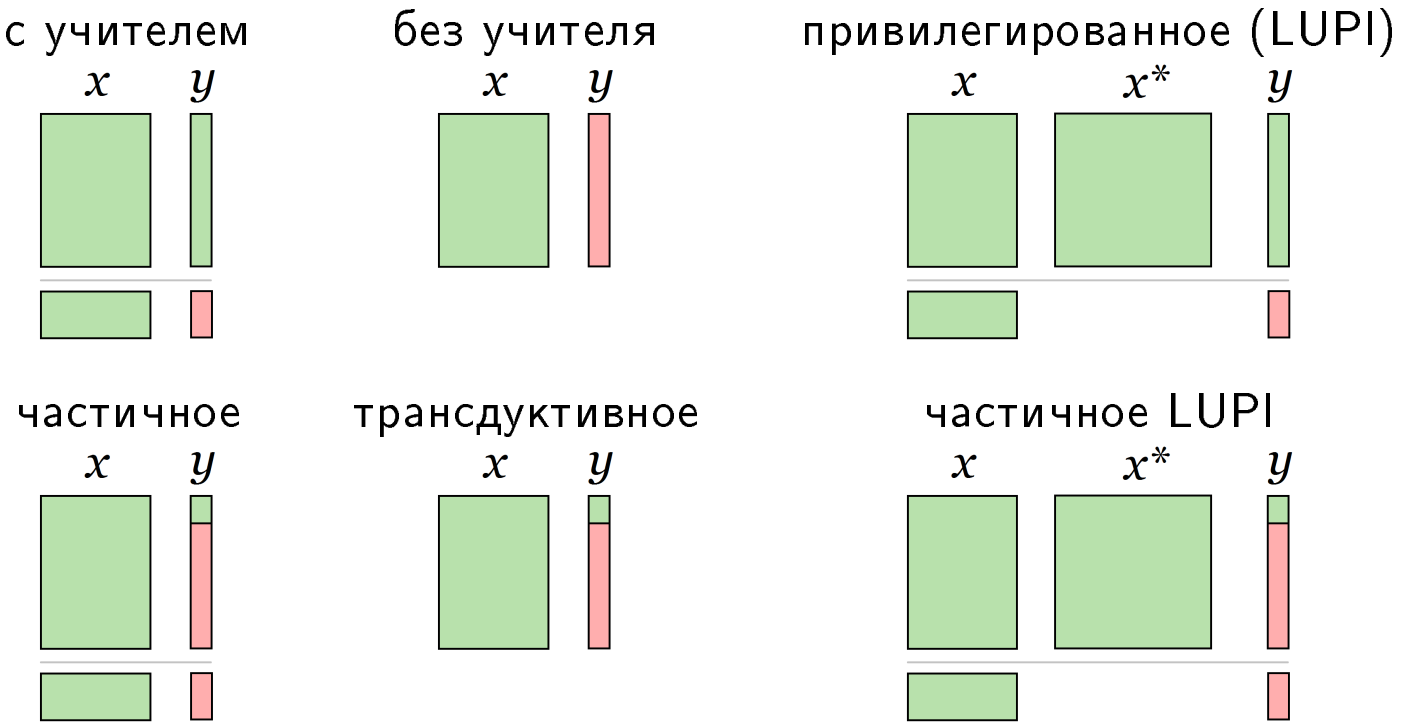
$$\sum_{i=1}^{\ell} \mathscr{L}\big(a(x_i, w), y_i\big) + \lambda \mathscr{L}\big(a(x_i^*, w^*), y_i\big) + \\ + \mu \mathscr{L}\big(a(x_i, w), a(x_i^*, w^*)\big) \to \min_{w, w*}$$

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

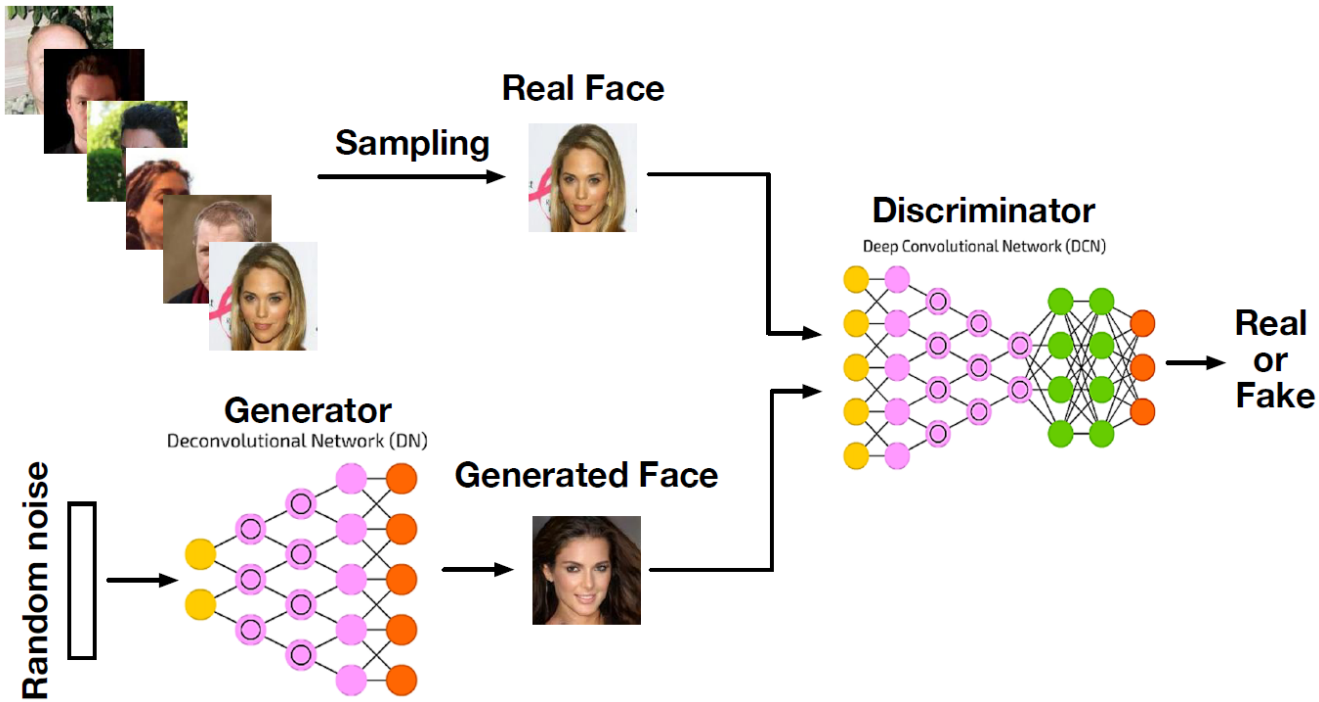
Варианты LUPI (Learning Using Priveleged Information):



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

Генеративная состязательная сеть (Generative Adversarial Net)

Генератор $G(z)$ учится порождать объекты x из шума z
Дискриминатор $D(x)$ учится отличать их от реальных объектов



Antonia Creswell et al. Generative Adversarial Networks: an overview. 2017.
Zhengwei Wang et al. Generative Adversarial Networks: a survey and taxonomy. 2019.
Chris Nicholson. A Beginner's Guide to Generative Adversarial Networks. https://pathmind.com/wiki/generative-adversarial-network-gan. 2019.

Постановка задачи GAN

Дано: выборка объектов $\{x_i\}_{i=1}^m$ из X

Найти:

вероятностную генеративную модель $G(z,\alpha)$: $x \sim p(x|z,\alpha)$

вероятностную дискриминативную модель $D(x,\beta) = p(1|x,\beta)$

Критерий:

обучение дискриминативной модели D:

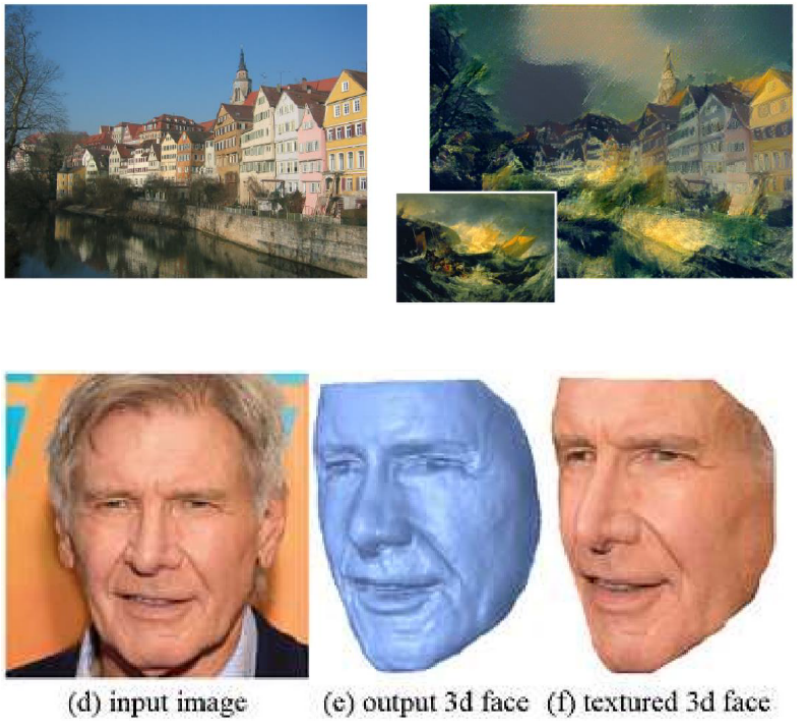
$$\sum_{i=1}^{m} \ln D(x_i,\beta) + \ln\big(1 - D\big(G(z_i,\alpha),\beta\big)\big) \;\to\; \max_{\beta}$$

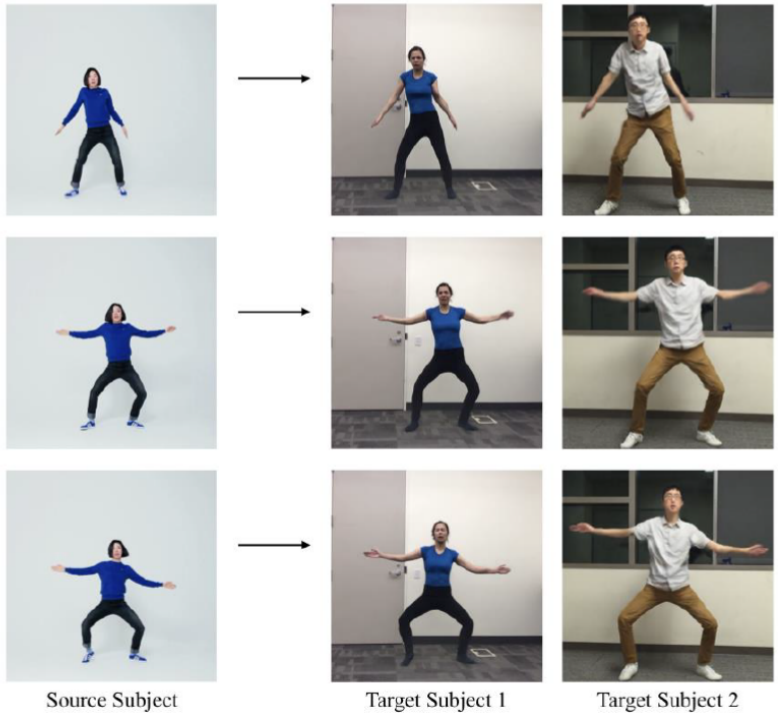
обучение генеративной модели G по случайному шуму $\{z_i\}_{i=1}^m$:

$$\sum_{i=1}^{m} \ln\big(1 - D\big(G(z_i,\alpha),\beta\big)\big) \;\to\; \min_{\alpha}$$

Ian Goodfellow et al. Generative Adversarial Nets. 2014

Примеры GAN для синтеза изображений и видео





Chuan Li, Michael Wand. Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks. 2016.

Xiaoxing Zeng, Xiaojiang Peng, Yu Qiao. DF2Net: A Dense Fine Finer Network for Detailed 3D Face Reconstruction. ICCV-2019.

Caroline Chan, Shiry Ginosar, Tinghui Zhou, Alexei A. Efros. Everybody Dance Now. ICCV-2019.

Вместо резюме. Типология задач машинного обучения

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

Вместо резюме. Типология задач машинного обучения

5. Обучение представлений (representation learning)
   - обучение признаков (feature learning)
   - матричные разложения (matrix factorization)
   - обучение многообразий (manifold learning)
6. Глубокое обучение (deep learning)
7. Обучение близости/связей (similarity/relational learning)
8. Обучение структуры модели (structure learning)
9. Привилегированное обучение (privileged learning, distilling)
10. Состязательное обучение (adversarial learning)
11. Динамическое обучение (online/incremental learning)
12. Активное обучение (active learning)
13. Обучение с подкреплением (reinforcement learning)
14. Перенос обучения (transfer learning)
15. Многозадачное обучение (multitask learning)
16. Мета-обучение (meta-learning, AutoML)