• Введение в машинное обучение •

Минимизация эмпирического риска, линейные модели и регуляризация

Воронцов Константин Вячеславович
k.v.vorontsov@phystech.edu
http://www.MachineLearning.ru/wiki?title=User:Vokov

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

МФТИ.ФПМИ.ИС.ИАД • 19 февраля 2026

Резюме предыдущей лекции

Машинное обучение — автоматизация научного метода

Задача машинного обучения начинается с её постановки

- ДНК задачи — Дано, Найти, Критерий

Основные понятия машинного обучения:

- объект, ответ, признак
- модель, функция потерь, эмпирический риск
- алгоритм обучения, недообучение, переобучение

Три главных принципа машинного обучения:

- эмпирическая индукция Фрэнсиса Бэкона
- минимизация (+регуляризация) эмпирического риска
- обучаемая векторизация данных (Deep Learning)

Содержание

Общая постановка большинства задач машинного обучения

Дано: X — пространство *объектов*
$X^\ell = \{x_1, \ldots, x_\ell\} \subset X$ — *обучающая выборка* (training sample)
$a(x, w)$, $a\colon X \times W \to Y$ — параметрическая модель, гипотеза

Найти $w \in W \subseteq \mathbb{R}^N$ — вектор параметров модели $a(x, w)$

Критерий min регуляризованного эмпирического риска
(Regularized ERM — Empirical Risk Minimization):

$$Q(w, X^\ell) = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathscr{L}(w, x_i) + \tau \mathscr{R}(w) \to \min_w$$

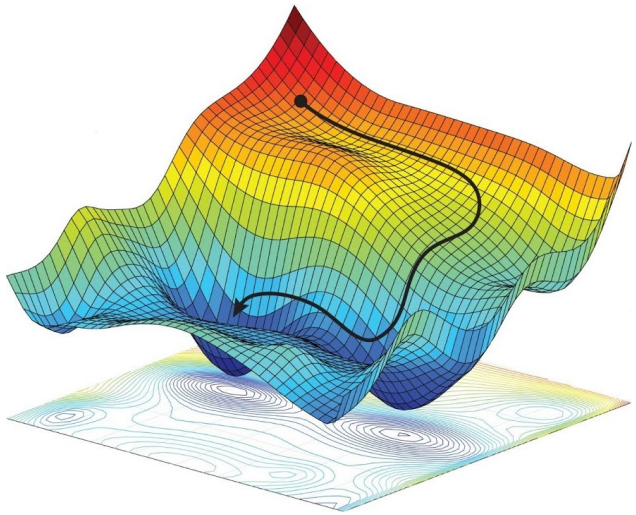
$\mathscr{L}(w, x)$ — *функция потерь* (loss function),
тем больше, чем хуже ответ модели $a(x, w)$ на объекте x

$\mathscr{R}(w)$ — *регуляризатор*, не прецедентные требования к модели

Вопрос: зачем нужен коэффициент регуляризации τ?

Градиентный метод минимизации эмпирического риска

$$Q(w) = \frac{1}{\ell}\sum_{i=1}^{\ell}\mathscr{L}(w, x_i) + \tau\mathscr{R}(w) \to \min_w$$



Метод *градиентного спуска*:

$w^{(0)} :=$ начальное приближение;

$w^{(t+1)} := w^{(t)} - h\nabla Q(w^{(t)})$

где $\nabla Q(w) = \left(\frac{\partial Q(w)}{\partial w_j}\right)_{j=1}^{N}$ — *вектор градиента*,

h — *градиентный шаг*, называемый также *темпом обучения*

$$w^{(t+1)} := w^{(t)} - h\left(\frac{1}{\ell}\sum_{i=1}^{\ell}\nabla\mathscr{L}\left(w^{(t)}, x_i\right) + \tau\nabla\mathscr{R}\left(w^{(t)}\right)\right)$$

Идея ускорения сходимости: брать случайное подмножество слагаемых, или вообще один объект, чаще обновляя вектор w

Метод стохастического градиента SG (Stochastic Gradient)

$$Q(w) = \frac{1}{\ell}\sum_{i=1}^{\ell}\mathscr{L}(w, x_i) + \tau\mathscr{R}(w) \to \min_w$$

Вход: выборка X^ℓ, параметры h, τ, λ;
Выход: вектор параметров w;
инициализировать параметры w_j, $j = 1, \ldots, N$;
инициализировать оценку $Q(w)$ по небольшой подвыборке;
повторять
  объект x_i выбрать из X^ℓ случайным образом;
  потеря: $\mathscr{L}_i := \mathscr{L}(w, x_i)$;
  градиентный шаг: $w := w - h\nabla\mathscr{L}(w, x_i) - h\tau\nabla\mathscr{R}(w)$;
  рекуррентная оценка критерия: $Q := \lambda\mathscr{L}_i + (1-\lambda)Q$;
пока значение Q и/или параметры w не сойдутся;

В: какие бывают условия остановки в методах оптимизации?

H. Robbins, S. Monro. A stochastic approximation method. 1951.

Откуда взялась такая рекуррентная оценка функционала?

Проблема: вычисление оценки Q по всей выборке $x_1, \ldots, x_\ell$ намного дольше градиентного шага по одному объекту x_i.

Решение: использовать приближённую рекуррентную формулу.

Среднее арифметическое:

$$\bar{Q}_m = \tfrac{1}{m}\mathscr{L}_m + \tfrac{1}{m}\mathscr{L}_{m-1} + \tfrac{1}{m}\mathscr{L}_{m-2} + \ldots$$

$$\bar{Q}_m = \tfrac{1}{m}\mathscr{L}_m + (1 - \tfrac{1}{m})\bar{Q}_{m-1}$$

Экспоненциальное скользящее среднее (ЭСС):

$$\bar{Q}_m = \lambda\mathscr{L}_m + (1-\lambda)\lambda\mathscr{L}_{m-1} + (1-\lambda)^2\lambda\mathscr{L}_{m-2} + \ldots$$

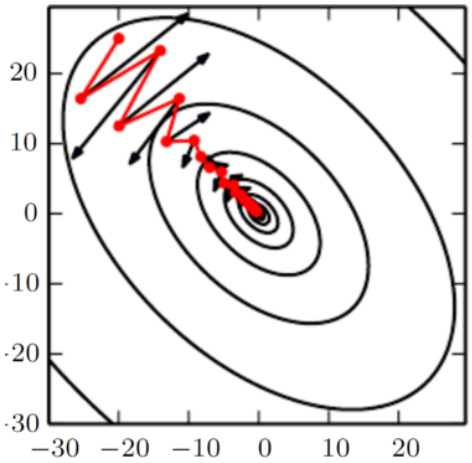
$$\bar{Q}_m = \lambda\mathscr{L}_m + (1 - \lambda)\bar{Q}_{m-1}$$

Параметр λ (порядка $\frac{1}{m}$) — *темп забывания* предыстории ряда.

Метод накопления инерции (momentum)

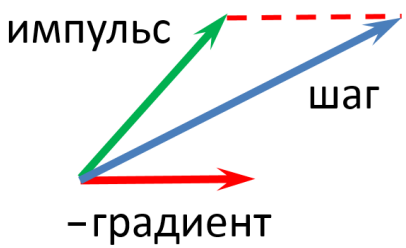
Momentum — экспоненциальное скользящее среднее градиента по последним $\approx \frac{1}{1-\gamma}$ итерациям [Б.Т.Поляк, 1964]:

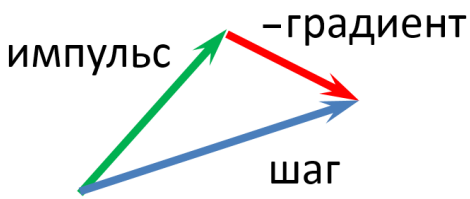
$$v := \gamma v + (1-\gamma)\nabla \mathscr{L}(w, x_i)$$
$$w := w - hv$$



NAG (Nesterov's accelerated gradient) — стохастический градиент с инерцией [Ю.Е.Нестеров, 1983]:

$$v := \gamma v + (1-\gamma)\nabla \mathscr{L}(w - h\gamma v, x_i)$$
$$w := w - hv$$





Эвристики (универсальные, не зависящие от вида модели)

1. **инициализация** w**:** случайная, быстрое приближённое решение по малой подвыборке, их комбинация, ...
2. **порядок предъявления объектов:**
   чем больше $\mathscr{L}_i$, тем выше вероятность снова выбрать x_i
3. **выбор градиентного шага:** *метод скорейшего спуска*
   основан на поиске оптимального *адаптивного шага* h^*:
   $$\mathscr{L}\big(w - h\nabla\mathscr{L}(w, x_i), x_i\big) \to \min_h$$
4. **методы второго порядка:** *метод Ньютона-Рафсона*
   $w := w - h\big(\mathscr{L}''(w, x_i)\big)^{-1}\nabla\mathscr{L}(w, x_i)$, $\mathscr{L}''$ — матрица Гессе
   диагональный метод Левенберга-Марквардта:
   обнулить все недиагональные элементы матрицы Гессе
5. **мультистарт:** многократные запуски из разных случайных начальных приближений и выбор лучшего решения

Метод SG: достоинства и недостатки

Достоинства:

1. *универсальность:* для любых $a(x, w)$, $\mathscr{L}(w, x)$, $\mathscr{R}(w)$
2. *простота:* относительно легко реализуется
3. *поточность:* возможность обучения на потоке данных
4. *подходит для обработки больших данных:*
   — можно получить неплохое решение, успев обработать лишь малую часть обучающей выборки
   — часто оказывается быстрее и лучше более сложных и ресурсоёмких методов второго порядка

Недостатки:

1. подбор комплекса эвристик является искусством (не забыть про переобучение, застревание, расходимость)

Задача обучения модели регрессии

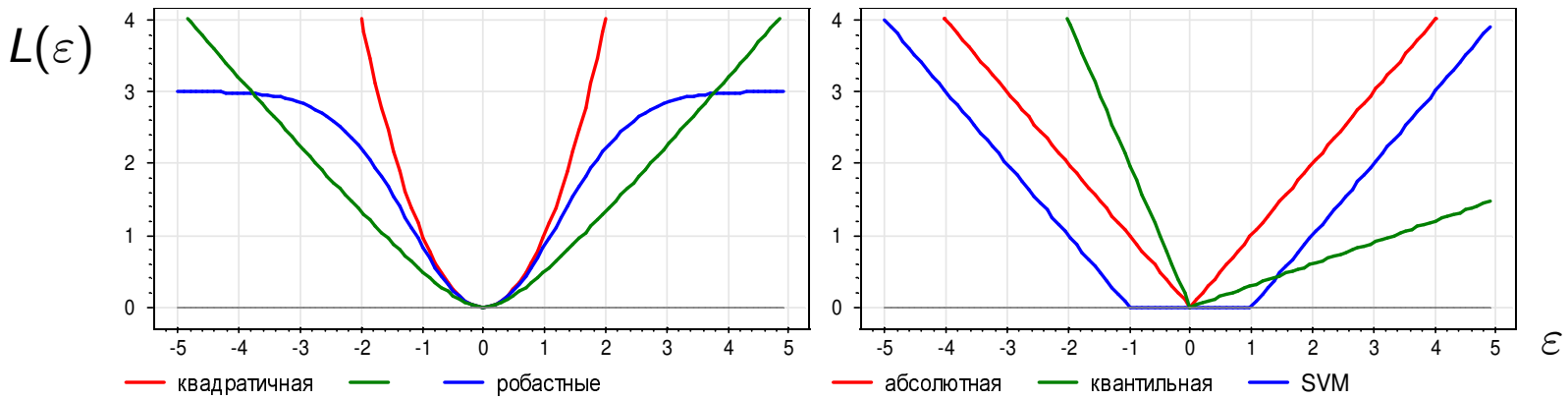
Дано: обучающая выборка $(x_i, y_i)_{i=1}^{\ell}$ с ответами $y_i \in \mathbb{R}$

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

Критерий: минимум эмпирического риска (ERM)

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

Унимодальные функции потерь $L(\varepsilon)$ от ошибки $\varepsilon = a(x, w) - y$:



В: какие свойства придаёт модели каждая из этих функций?

Пример. Задача прогнозирования ~~спроса~~ объёмов продаж

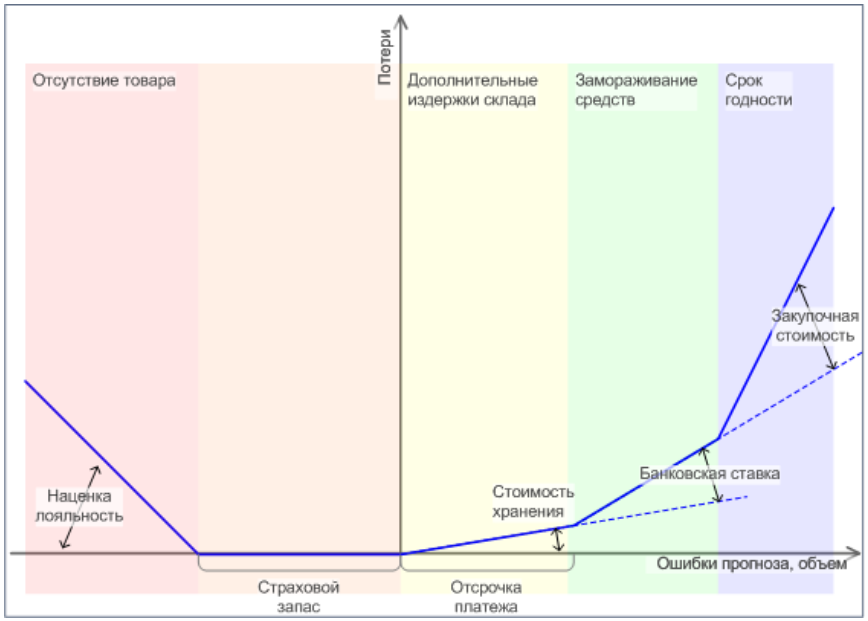
Объект — тройка $\langle$товар, магазин, день$\rangle$.

Примеры признаков:

- **бинарные:** выходной день, праздник, промоакция, и т. д.
- **количественные:** объёмы продаж в предшествующие дни.

Особенности задачи:

- функция потерь
  кусочно линейная,
  не квадратичная,
  не симметричная
- разреженные данные:
  если запасы нулевые,
  то и продажи нулевые



Задача обучения модели бинарной классификации

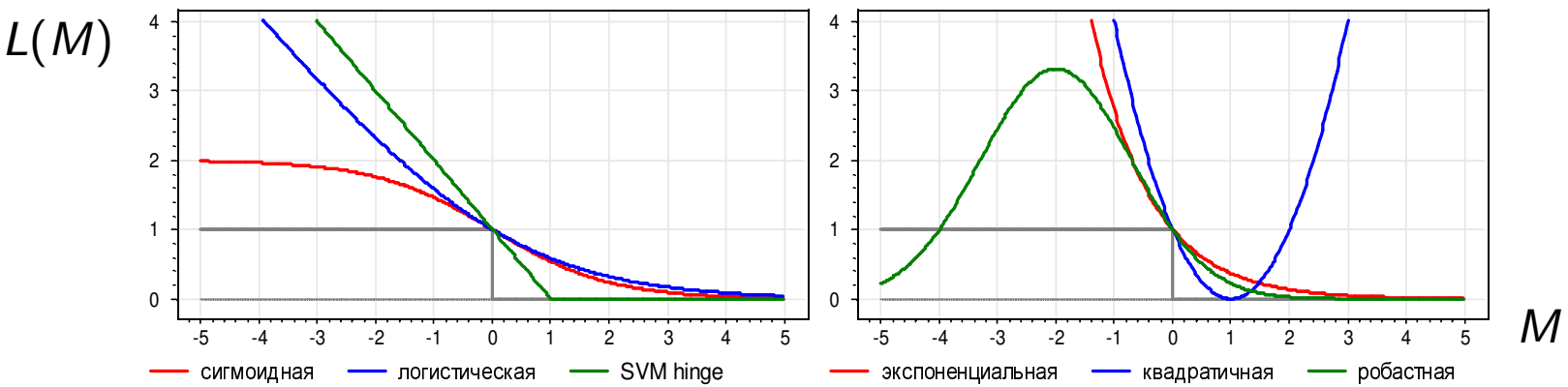
Дано: обучающая выборка $(x_i, y_i)_{i=1}^{\ell}$, $y_i \in \{-1, +1\}$

Найти: вектор w модели классификации $a(x, w) = \operatorname{sign} g(x, w)$

Критерий min *аппроксимированного* эмпирического риска:

$$\sum_{i=1}^{\ell} \big[g(x_i, w)y_i < 0\big] \leqslant \sum_{i=1}^{\ell} L\big(g(x_i, w)y_i\big) \;\to\; \min_w$$

Убывающие функции потерь $L(M)$ от *отступа* $M = g(x, w)y$:



В: какие свойства придаёт модели каждая из этих функций?

Бинарный разделяющий классификатор (margin-based classifier)

Бинарный классификатор: $a(x, w) = \operatorname{sign} g(x, w), \quad Y = \{-1, +1\}$

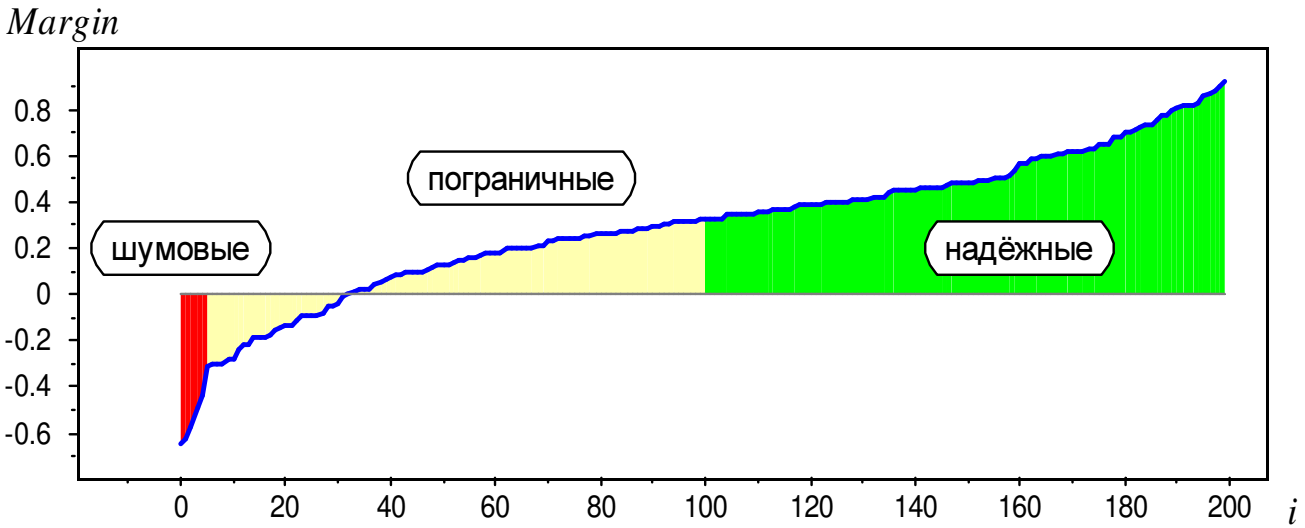
$g(x, w)$ — *разделяющая (дискриминантная) функция*

$\{x : g(x, w) = 0\}$ — разделяющая поверхность между классами

$M_i(w) = g(x_i, w)y_i$ — *отступ* (margin) объекта x_i

$M_i(w) < 0 \iff$ модель $a(x, w)$ ошибается на x_i

Ранжирование объектов по возрастанию отступов $M_i(w)$:



Задача обучения модели многоклассовой классификации

Дано: обучающая выборка $(x_i, y_i)_{i=1}^{\ell}$, $y_i \in Y$, $|Y| < \infty$

Найти: вектор параметров $w = (w_y : y \in Y)$ модели:

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

Критерий «каждый против всех остальных» (One-vs.-All):
вводится $M_{iy}(w)$ — отступ объекта x_i относительно класса y

$$\sum_{i=1}^{\ell} \sum_{z \neq y_i} \Big[\underbrace{g_{y_i}(x_i, w_{y_i}) - g_z(x_i, w_z)}_{M_{iz}(w)} < 0\Big] \leqslant \sum_{i=1}^{\ell} \sum_{z \neq y_i} L\big(M_{iz}(w)\big) \to \min_w$$

$M_i(w) = \min\limits_{z \neq y_i} M_{iz}(w)$ — *многоклассовый отступ* объекта x_i
(годится для оценивания, но критерий на нём строить нельзя)

Y.Wang, C.Scott. Unified binary and multiclass margin-based classification. 2024

Задача обучения модели ранжирования

Дано: обучающая выборка $(x_1, \ldots, x_\ell)$,
$i \prec j$ — отношение «x_j лучше, чем x_i» между объектами из X^ℓ

Найти: параметры w модели ранжирования $a(x, w)$, которая аппроксимирует отношение порядка, заданное на выборке:

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

Критерий: число неверно ранжированных пар объектов

$$Q(w) = \sum_{i \prec j} \big[\underbrace{a(x_j, w) - a(x_i, w)}_{M_{ij}(w)} < 0 \big] \leqslant$$

$$\leqslant \sum_{i \prec j} L\big(a(x_j, w) - a(x_i, w)\big) \ \to \ \min_w$$

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

Идея SG: градиентные шаги по парам объектов x_i, x_j: $i \prec j$

Примеры задач ранжирования

Ранжирование (Learning to Rank, LtR, L2R, LETOR) нужно в системах человеко-машинного принятия решений, когда **машина** формирует ранжированный список вариантов, **человек** (ответственно) выбирает и принимает решение

- ранжирование выдачи поисковой системы
- ранжирование рекомендаций пользователям (книги, фильмы, музыка, товары интернет-магазина, и т.п.)
- ранжирование вариантов автоматического завершения запроса (Query Auto Completion, auto-suggest)
- ранжирование возможных ответов в диалоговых системах (Question Answering Systems)
- ранжирование вариантов перевода в системах машинного перевода (Machine Translation)

Логистическая регрессия для бинарной классификации

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

Найти: параметры $w \in \mathbb{R}^n$ линейной модели классификации

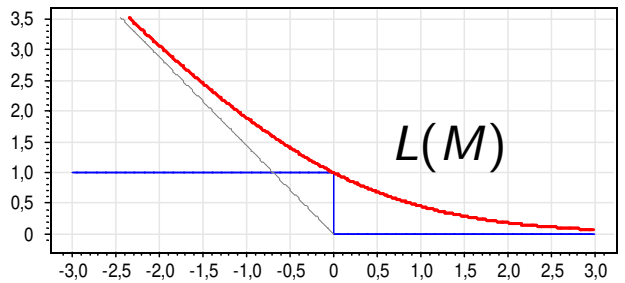
$$a(x, w) = \text{sign} \sum_{j=1}^{n} w_j f_j(x) = \text{sign} \langle w, x \rangle$$

Критерий min *аппроксимированного эмпирического риска:*

$$\sum_{i=1}^{\ell} \ln\left(1 + \exp(-\underbrace{\langle w, x_i \rangle y_i}_{\text{margin}})\right) \to \min_w$$

логарифмическая функция потерь:

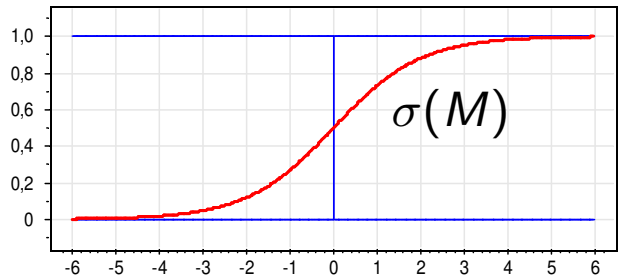
$$L(M) = \ln\left(1 + e^{-M}\right)$$



Условные вероятности классов:

$$P(y|x, w) = \sigma(\langle w, x \rangle y)$$

$\sigma(M) = \frac{1}{1+e^{-M}}$ — сигмоидная функция



Мультиколлинеарность и переобучение в линейных моделях

$a(x, w) = \langle w, x \rangle$ — линейная модель регрессии
$a(x, w) = \text{sign}\langle w, x \rangle$ — линейная модель классификации

Мультиколлинеарность — линейная зависимость признаков:

- $\exists u \in \mathbb{R}^n$: $\forall x \in X \;\; \langle u, x \rangle = 0$
- неединственность решения: $\forall \gamma \in \mathbb{R} \;\; \langle w, x \rangle = \langle w + \gamma u, x \rangle$

Мультиколлинеарность приводит к переобучению:

- неустойчивость: слишком большие веса $|w_j|$ разных знаков
- переобучение: $Q(w^*, X^\ell) \ll Q(w^*, X^k)$

Как уменьшить переобучение:

- регуляризация $\|w\| \to \min$ (сокращение весов, weight decay)
- отбор признаков: $f_1, \ldots, f_n \to f_{j_1}, \ldots, f_{j_m}, \;\; m \ll n$
- преобразование признаков: $f_1, \ldots, f_n \to g_1, \ldots, g_m, \;\; m \ll n$

Регуляризация по L_2-норме (сокращение весов, weight decay)

Штраф за увеличение нормы вектора весов:

$$\mathscr{L}_\tau(w, x_i) = \mathscr{L}(w, x_i) + \frac{\tau}{2}\|w\|^2 = \mathscr{L}(w, x_i) + \frac{\tau}{2}\sum_{j=1}^{n} w_j^2 \to \min_w .$$

Градиент:

$$\nabla\mathscr{L}_\tau(w, x_i) = \nabla\mathscr{L}(w, x_i) + \tau w.$$

Модификация градиентного шага в методе SG:

$$w := w(1 - h\tau) - h\nabla\mathscr{L}(w, x_i).$$

Методы подбора коэффициента регуляризации τ:

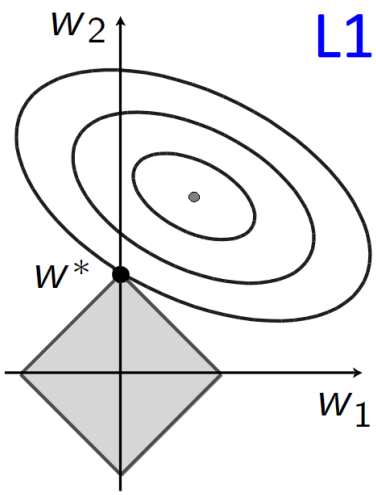
1. hold-out или скользящий контроль
2. стохастическая адаптация по сетке значений $\{\tau_k\}$

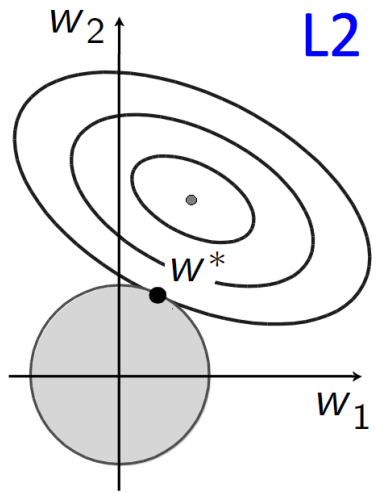
Регуляризация по L_p-норме ($p = 1, 2, \ldots$) для отбора признаков

LASSO — Least Absolute Shrinkage and Selection Operator

$$Q(w) + \tau \sum_{j=1}^{n} |w_j|^p \to \min_w \quad \Longleftrightarrow \quad \begin{cases} Q(w) \to \min\limits_w; \\ \sum\limits_{j=1}^{n} |w_j|^p \leqslant \varkappa; \end{cases}$$

$$\sum_{j=1}^{n} |w_j| \leqslant \varkappa$$





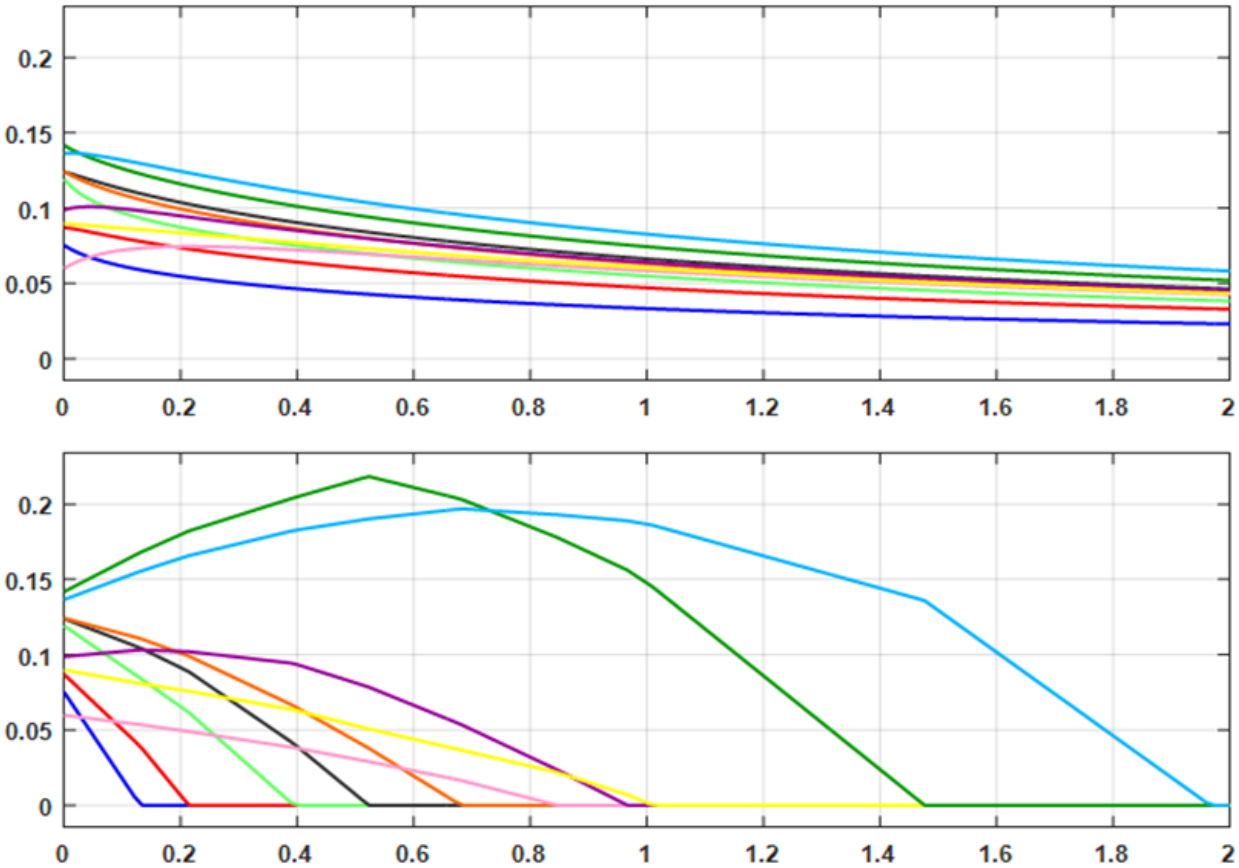
$$\sum_{j=1}^{n} w_j^2 \leqslant \varkappa$$

В: как доказать эквивалентность этих двух постановок?

T.Hastie, R.Tibshirani, J.Friedman. The Elements of Statistical Learning. 2017.

Сравнение L_2 и L_1 регуляризации

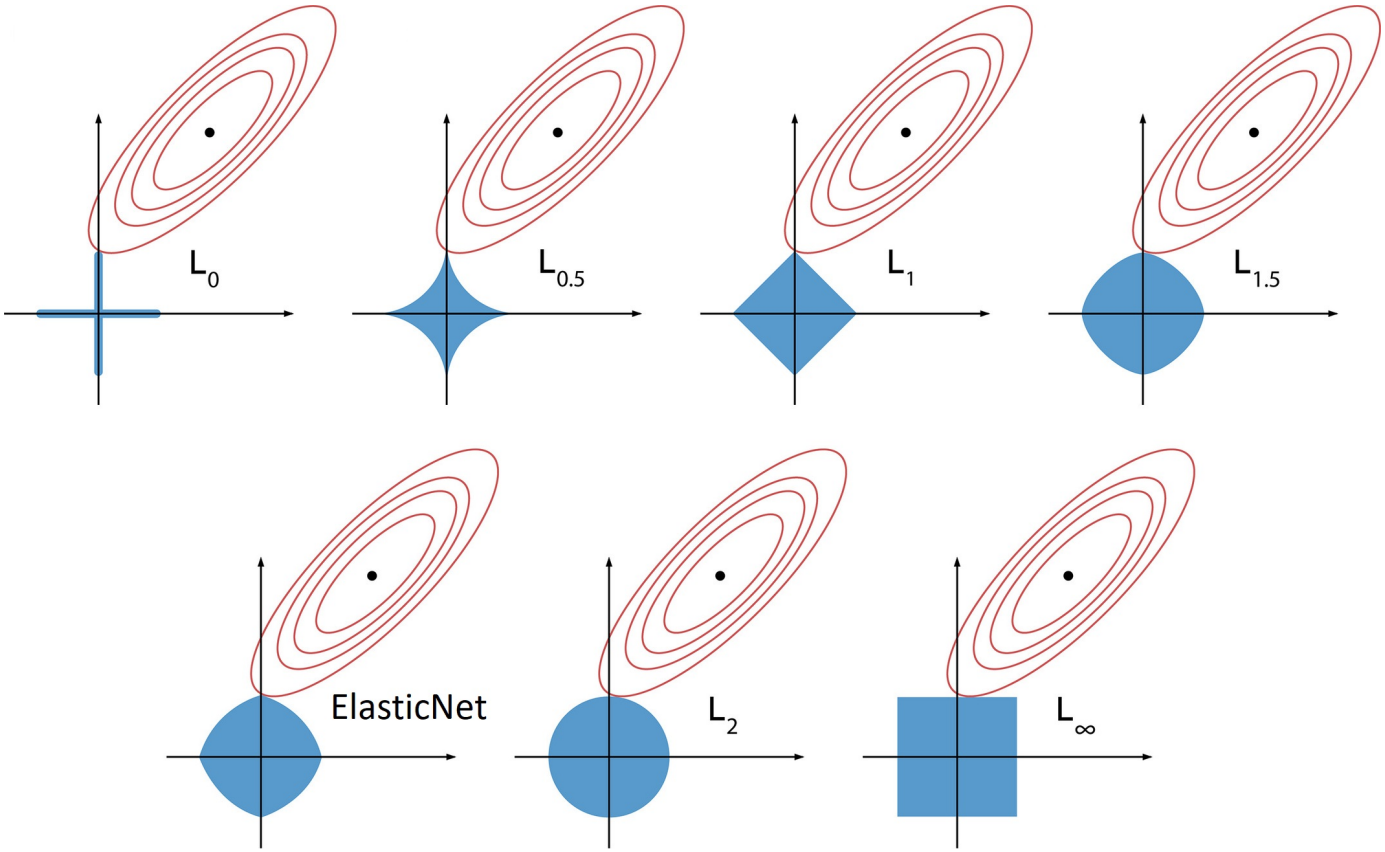
Зависимость весов w_j от параметра селективности τ



LASSO усиливает отбор признаков по мере увеличения τ

Геометрическая интерпретация отбора признаков

Сравнение регуляризаторов по различным L_p-нормам:



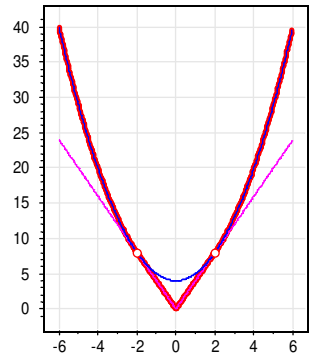
Негладкость регуляризатора приводит к отбору признаков

Негладкие регуляризаторы для отбора и группировки признаков

Общий вид регуляризаторов (τ — параметр селективности):

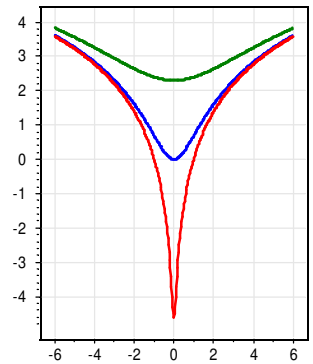
$$\sum_{i=1}^{\ell} \mathscr{L}(w, x_i) + \sum_{j=1}^{n} R_\tau(w_j) \;\to\; \min_w .$$

LASSO (L_1): $R_\tau(w) = \tau|w|$
— слишком агрессивно отбрасывает признаки



Elastic Net: $R_\tau(w) = \tau|w| + \lambda w^2$
— группирует информативные признаки

Support Feature Machine (SFM):
$R_\tau(w) = \big[|w| \leqslant \tau\big] \cdot 2\tau|w| + \big[|w| > \tau\big] \cdot (\tau^2 + w^2)$
— не группирует неинформативные признаки



Relevance Feature Machine (RFM):
$R_\tau(w) = \ln\left(\tau w^2 + 1\right)$
— лучше находит совместно значимые признаки

Метод опорных векторов (Support Vector Machine, SVM)

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

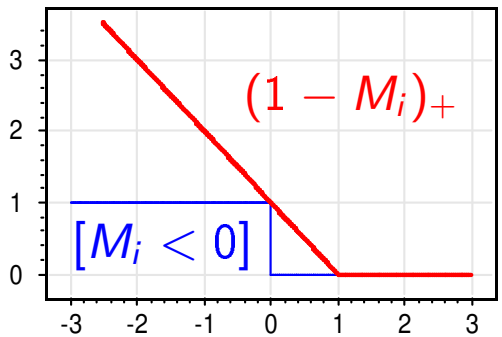
Найти: параметры $w \in \mathbb{R}^n$, $w_0 \in \mathbb{R}$ линейной модели

$$a(x; w, w_0) = \text{sign}\big(\langle x, w\rangle - w_0\big)$$

Критерий *аппроксимированного и регуляризированного* ERM:

$$\sum_{i=1}^{\ell} \Big(1 - \underbrace{\big(\langle x_i, w\rangle - w_0\big) y_i}_{\text{margin } M_i(w, w_0)}\Big)_+ + \frac{1}{2C}\|w\|^2 \;\to\; \min_{w, w_0}$$

- *Аппроксимация* штрафует объекты за приближение к границе классов, увеличивая зазор между классами
- *Регуляризация* штрафует неустойчивые решения в случае мультиколлинеарности

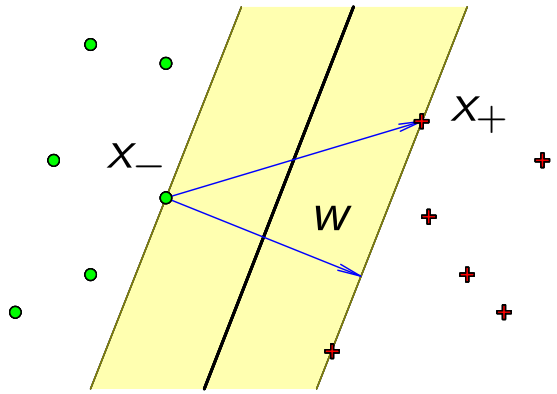


В.Н.Вапник, А.Я.Лернер. Узнавание образов при помощи обобщенных портретов. 1963
C.Cortes, V.Vapnik. Support vector networks. 1995.

Основные свойства метода опорных векторов

1. **Геометрия:** ширина зазора (разделяющей полосы) между классами максимальна,

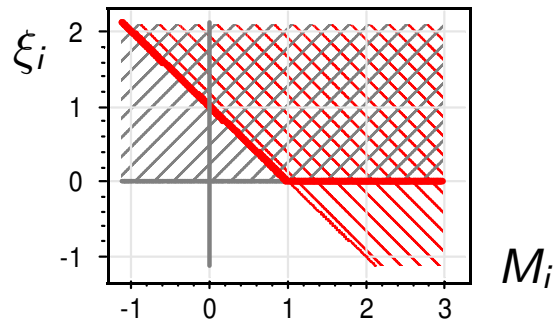
   разделяющая гиперплоскость проходит посередине полосы.



2. **Эквивалентная** гладкая задача выпуклого программирования с переменными $\xi_i = \big(1 - M_i(w, w_0)\big)_+$:

$$\begin{cases} \dfrac{1}{2}\|w\|^2 + C\sum\limits_{i=1}^{\ell} \xi_i \to \min\limits_{w, w_0, \xi} \\ \xi_i \geqslant 1 - M_i(w, w_0), \quad i = 1, \ldots, \ell \\ \xi_i \geqslant 0, \quad i = 1, \ldots, \ell \end{cases}$$

ξ_i — slack variable, штраф за нарушение разделяющей полосы



Основные свойства метода опорных векторов

3. **Двойственная задача** относительно переменных λ_i:

$$\begin{cases} \sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\lambda_i\lambda_j y_i y_j \langle x_i, x_j\rangle - 2\sum_{i=1}^{\ell}\lambda_i \;\to\; \min\limits_{\lambda}; \\ \sum_{i=1}^{\ell}\lambda_i y_i = 0; \qquad 0 \leqslant \lambda_i \leqslant C, \quad i = 1,\ldots,\ell \end{cases}$$

4. **Опорные векторы** — объекты x_i: $M_i \leqslant 1$, $\lambda_i > 0$, через которые линейно выражается вектор решения w:

$$w = \sum_{i=1}^{\ell}\lambda_i y_i x_i, \qquad w_0 = \langle w, x_i\rangle - y_i \;\text{ для } i\colon M_i = 1$$

5. **Разреженность решения:** опорными векторами является, как правило, небольшая доля объектов выборки

S.Fine, K.Scheinberg. INCAS: An incremental active set method for SVM. 2002.
J.Platt. Fast training support vector machines using sequential minimal optimization. 1999.

Основные свойства метода опорных векторов

6. **Скалярные произведения** $f_i(x) = \langle x, x_i \rangle$ с опорными x_i образуют новые признаки для линейного классификатора:

$$a(x; w, w_0) = \text{sign}\left(\sum_{i=1}^{\ell} \lambda_i y_i \langle x, x_i \rangle - w_0 \right)$$

7. **Подстановка ядра** (kernel trick) в двойственной задаче и в классификаторе делает модель нелинейной:

$$a(x; w, w_0) = \text{sign}\left(\sum_{i=1}^{\ell} \lambda_i y_i K(x, x_i) - w_0 \right)$$

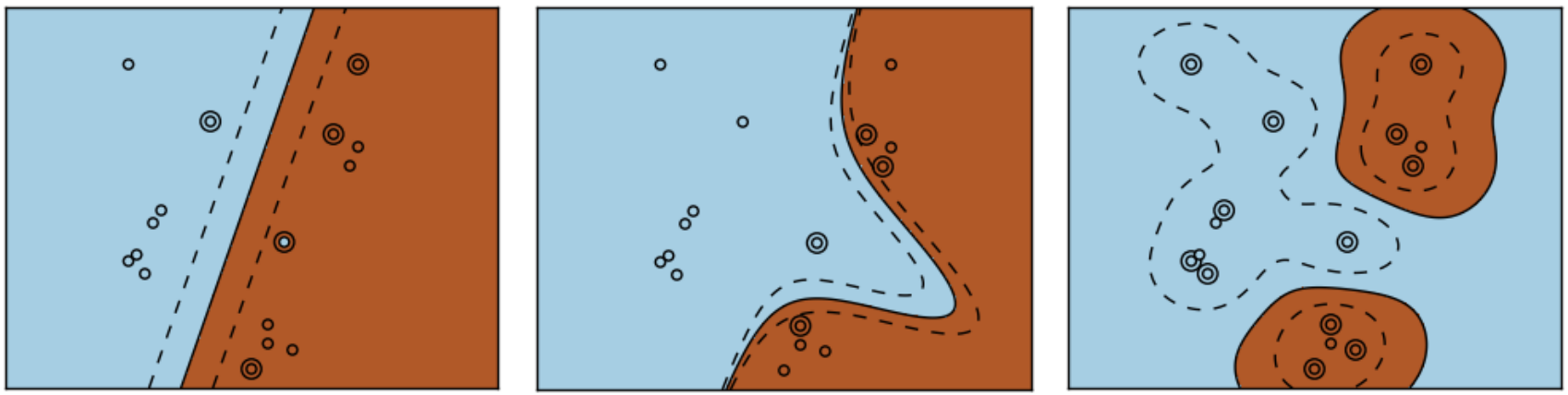
8. **Ядро** — скалярное произведение $K(x, x') = \langle \psi(x), \psi(x') \rangle_H$ в «спрямляющем» гильбертовом пространстве H

Классификация с различными ядрами

Гиперплоскость в спрямляющем пространстве соответствует нелинейной разделяющей поверхности в исходном.

Примеры с различными ядрами $K(x, x')$

линейное	полиномиальное	гауссовское (RBF)
$\langle x, x' \rangle$	$\big(\langle x, x' \rangle + 1\big)^d, \quad d{=}3$	$\exp\big(-\gamma \|x - x'\|^2\big)$



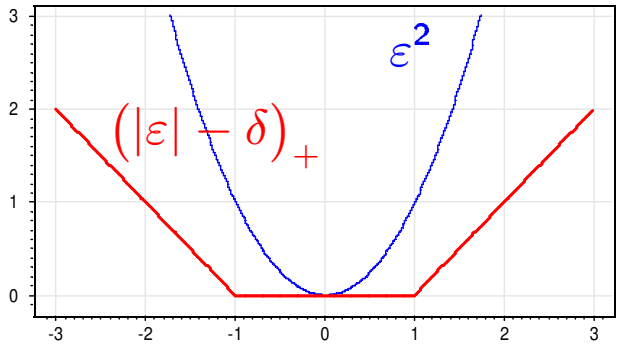
На практике C и K подбирают под задачу перебором

Пример из Python SkLearn: `http://scikit-learn.org/dev`

Задача обучения SVM-регрессии

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

Найти: параметры w и w_0 линейной модели регрессии $a(x, w) = \langle w, x \rangle - w_0$



$\varepsilon = a - y$

Критерий регуляризированного эмпирического риска:

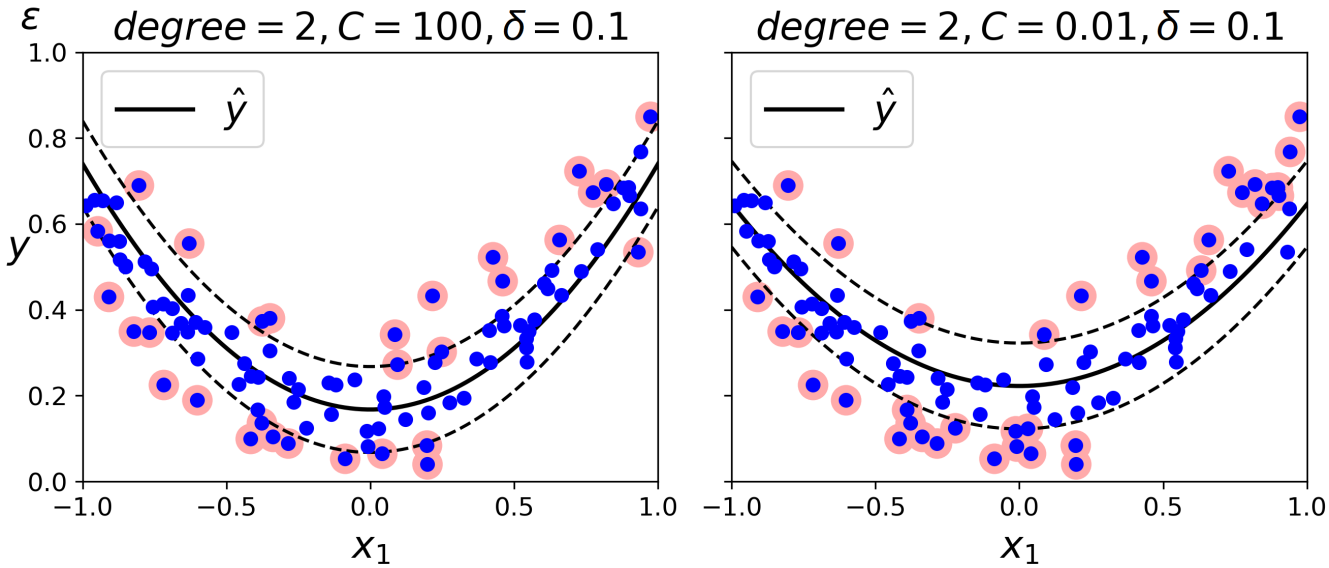
$$\sum_{i=1}^{\ell} \big(|\langle w, x_i \rangle - w_0 - y_i| - \delta\big)_+ + \frac{1}{2C}\|w\|^2 \to \min_{w, w_0}$$

Свойства, общие с SVM-классификацией:

- задача выпуклая, решение единственно
- решение разреженное, выражается через опорные векторы
- возможна замена $\langle x, x_i \rangle$ нелинейным ядром $K(x, x_i)$

SVM-регрессия. Пример 1

SVM-регрессия с полиномиальным ядром степени 2:

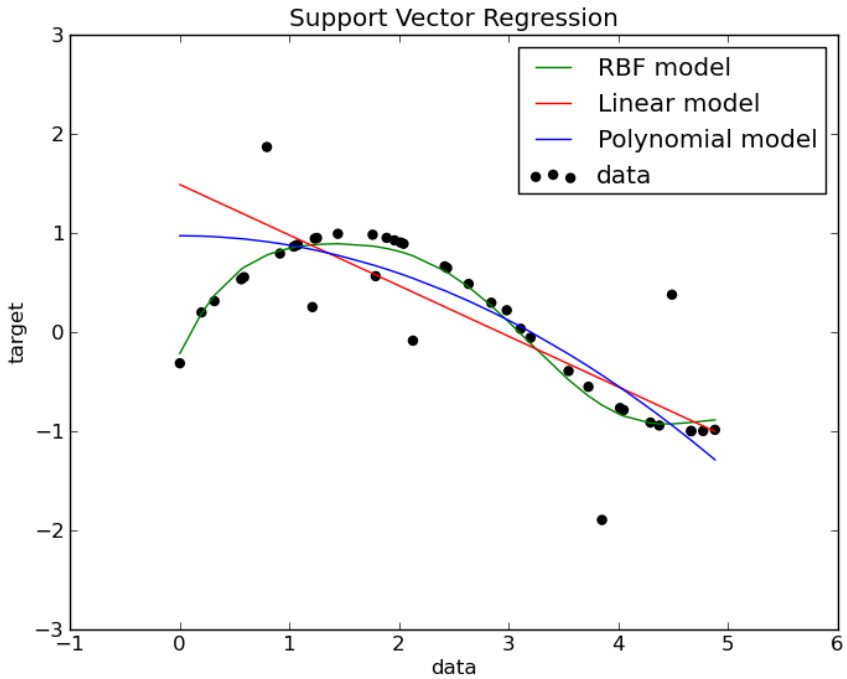


- Выделены опорные векторы
- Типично для SVM: результат обучения слабо зависит от коэффициента C, который подбирают по грубой сетке

http://scikit-learn.org/0.5/auto_examples/svm/plot_svm_regression.html

SVM-регрессия. Пример 2

Сравнение SVM-регрессии с гауссовским (RBF) ядром, линейной и полиномиальной регрессией:



- SVM-регрессия игнорирует выбросы (робастность)
- Удачный выбор ядра имеет значение!

---

http://scikit-learn.org/0.5/auto_examples/svm/plot_svm_regression.html

Резюме в конце лекции

- Метод стохастического градиента (SG) для ERM:
  — подходит для любых моделей и функций потерь
  — подходит для обучения по большим данным
- *Аппроксимация пороговой функции потерь* $L(M)$
  — непрерывная оптимизация для дискретных критериев
  — общий приём для классификации и ранжирования
- *Регуляризация* — в линейных моделях снижает переобучение, возникающее из-за мультиколлинеарности
- *Логистическая регрессия* — метод классификации, оценивающий условные вероятности классов $P(y|x)$
- *Метод опорных векторов* — изящно обобщается ядрами для нелинейной классификации и регрессии
- *Негладкость функции потерь* $\Rightarrow$ отбор объектов (SVM)
- *Негладкость регуляризатора* $\Rightarrow$ отбор признаков (LASSO)