Задача обучения линейного классификатора

Дано:

Обучающая выборка $X^\ell = (x_i, y_i)_{i=1}^\ell$,

x_i — объекты, векторы из множества $X = \mathbb{R}^n$,

y_i — метки классов, элементы множества $Y = \{-1, +1\}$.

Найти:

Параметры $w \in \mathbb{R}^n$, $w_0 \in \mathbb{R}$ линейной модели классификации

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

Критерий — минимизация эмпирического риска:

$$\sum_{i=1}^{\ell} \big[a(x_i; w, w_0) \neq y_i\big] = \sum_{i=1}^{\ell} \big[M_i(w, w_0) < 0\big] \;\to\; \min_{w, w_0}.$$

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

$b(x) = \langle x, w\rangle - w_0$ — *дискриминантная функция*.

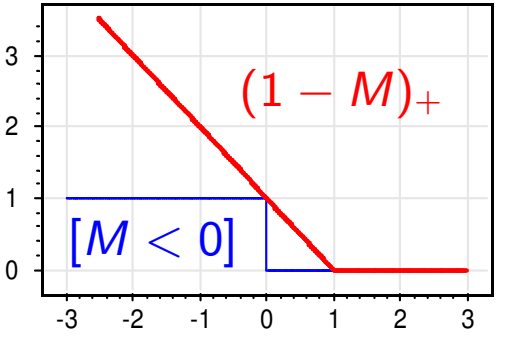
Аппроксимация и регуляризация эмпирического риска

Эмпирический риск — это кусочно-постоянная функция.
Заменим его оценкой сверху, непрерывной по параметрам:

$$Q(w, w_0) = \sum_{i=1}^{\ell} \big[M_i(w, w_0) < 0\big] \leqslant$$

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

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



Оптимальная разделяющая гиперплоскость

Линейный классификатор:

$$a(x, w) = \operatorname{sign}\big(\langle w, x\rangle - w_0\big), \quad w, x \in \mathbb{R}^n,\ w_0 \in \mathbb{R}.$$

Пусть выборка $X^\ell = (x_i, y_i)_{i=1}^\ell$ линейно разделима:

$$\exists w, w_0 : \quad M_i(w, w_0) = y_i\big(\langle w, x_i\rangle - w_0\big) > 0, \quad i = 1, \ldots, \ell$$

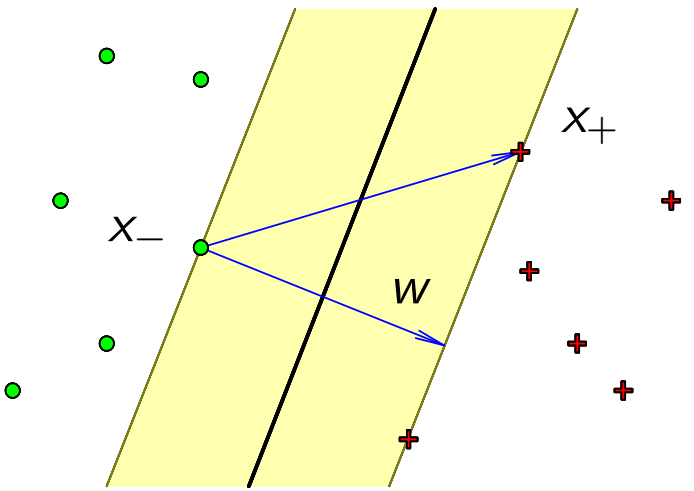
Нормировка: $\min\limits_{i=1,\ldots,\ell} M_i(w, w_0) = 1.$

Разделяющая полоса:

$$\big\{x : \ -1 \leqslant \langle w, x\rangle - w_0 \leqslant 1\big\}.$$

Ширина полосы:

$$\frac{\langle x_+ - x_-, w\rangle}{\|w\|} = \frac{2}{\|w\|} \to \max.$$



Переход к линейно неразделимой выборке

Постановка задачи в случае линейно разделимой выборки:

$$\begin{cases} \frac{1}{2}\|w\|^2 \to \min\limits_{w,w_0}; \\ M_i(w,w_0) \geqslant 1, \quad i = 1,\ldots,\ell. \end{cases}$$

Общий случай — линейно неразделимая выборка:

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

Исключая ξ_i, получаем задачу безусловной минимизации:

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

Условия Каруша–Куна–Таккера

Задача математического программирования:

$$\begin{cases} f(x) \to \min\limits_{x}; \\ g_i(x) \leqslant 0, \quad i = 1, \ldots, m; \\ h_j(x) = 0, \quad j = 1, \ldots, k. \end{cases}$$

Необходимые условия. Если x — точка локального минимума, то существуют множители μ_i, $i = 1, \ldots, m$, λ_j, $j = 1, \ldots, k$:

$$\begin{cases} \dfrac{\partial \mathscr{L}}{\partial x} = 0, \quad \mathscr{L}(x; \mu, \lambda) = f(x) + \sum\limits_{i=1}^{m} \mu_i g_i(x) + \sum\limits_{j=1}^{k} \lambda_j h_j(x); \\ g_i(x) \leqslant 0; \ h_j(x) = 0; \ \text{(исходные ограничения)} \\ \mu_i \geqslant 0; \ \text{(двойственные ограничения)} \\ \mu_i g_i(x) = 0; \ \text{(условие дополняющей нежёсткости)} \end{cases}$$

Применение условий KKT к задаче SVM

Функция Лагранжа:

$$\mathscr{L}(w, w_0, \xi; \lambda, \eta) =$$

$$= \frac{1}{2}\|w\|^2 - \sum_{i=1}^{\ell} \lambda_i \big(M_i(w, w_0) - 1\big) - \sum_{i=1}^{\ell} \xi_i \big(\lambda_i + \eta_i - C\big),$$

λ_i — переменные, двойственные к ограничениям $M_i \geqslant 1 - \xi_i$;
η_i — переменные, двойственные к ограничениям $\xi_i \geqslant 0$.

$$\begin{cases} \dfrac{\partial \mathscr{L}}{\partial w} = 0, \quad \dfrac{\partial \mathscr{L}}{\partial w_0} = 0, \quad \dfrac{\partial \mathscr{L}}{\partial \xi} = 0; \\ \xi_i \geqslant 0, \quad \lambda_i \geqslant 0, \quad \eta_i \geqslant 0, \quad i = 1, \ldots, \ell; \\ \lambda_i = 0 \;\text{ либо }\; M_i(w, w_0) = 1 - \xi_i, \quad i = 1, \ldots, \ell; \\ \eta_i = 0 \;\text{ либо }\; \xi_i = 0, \quad i = 1, \ldots, \ell; \end{cases}$$

Двойственная задача

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

Решив эту задачу численно относительно λ_i,
получаем линейный классификатор:

$$a(x) = \operatorname{sign}\Big(\sum_{i=1}^{\ell} \lambda_i y_i \langle x_i, x \rangle - w_0 \Big),$$

где $w_0 = \sum\limits_{i=1}^{\ell} \lambda_i y_i \langle x_i, x_j \rangle - y_j$ для такого j, что $\lambda_j > 0$, $M_j = 1$

Определение

Объект x_i называется *опорным*, если $\lambda_i \neq 0$.

Преимущества и недостатки SVM

Преимущества:

- Задача выпуклого квадратичного программирования имеет единственное решение.
- Выделяется множество опорных объектов.
- Имеются эффективные численные методы для SVM.
- Изящное обобщение на нелинейные классификаторы.

Недостатки:

- Опорными объектами могут становиться выбросы.
- Нет отбора признаков в исходном пространстве X.
- Приходится подбирать константу C.

Двойственная задача

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

Решив эту задачу численно относительно λ_i,
получаем линейный классификатор:

$$a(x) = \operatorname{sign}\Big(\sum_{i=1}^{\ell} \lambda_i y_i \langle x_i, x \rangle - w_0 \Big),$$

где $w_0 = \sum\limits_{i=1}^{\ell} \lambda_i y_i \langle x_i, x_j \rangle - y_j$ для такого j, что $\lambda_j > 0$, $M_j = 1$

Определение

Объект x_i называется *опорным*, если $\lambda_i \neq 0$.

Двойственная задача: нелинейное обобщение SVM

$$\begin{cases} -\sum\limits_{i=1}^{\ell}\lambda_i + \frac{1}{2}\sum\limits_{i=1}^{\ell}\sum\limits_{j=1}^{\ell}\lambda_i\lambda_j y_i y_j K(x_i, x_j) \;\to\; \min\limits_{\lambda}; \\ 0 \leqslant \lambda_i \leqslant C, \quad i = 1, \ldots, \ell; \\ \sum\limits_{i=1}^{\ell}\lambda_i y_i = 0. \end{cases}$$

Решив эту задачу численно относительно λ_i,
получаем линейный классификатор:

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

где $w_0 = \sum\limits_{i=1}^{\ell}\lambda_i y_i K(x_i, x_j) - y_j$ для такого j, что $\lambda_j > 0$, $M_j = 1$

Определение

Объект x_i называется *опорным*, если $\lambda_i \neq 0$.

Ядра для нелинейного обобщения SVM

Определение

Функция от пары объектов $K(x, x')$ называется *ядром*, если она представима в виде скалярного произведения

$$K(x, x') = \langle \psi(x), \psi(x') \rangle$$

при некотором преобразовании $\psi: X \to H$ из пространства признаков X в новое *спрямляющее* пространство H.

Возможная интерпретация:
признак $f_i(x) = K(x_i, x)$ — это оценка близости объекта x к опорному объекту x_i. Выбирая опорные объектов, SVM осуществляет отбор признаков в линейном классификаторе

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

Примеры ядер

Ядра в SVM расширяют линейную модель классификации:

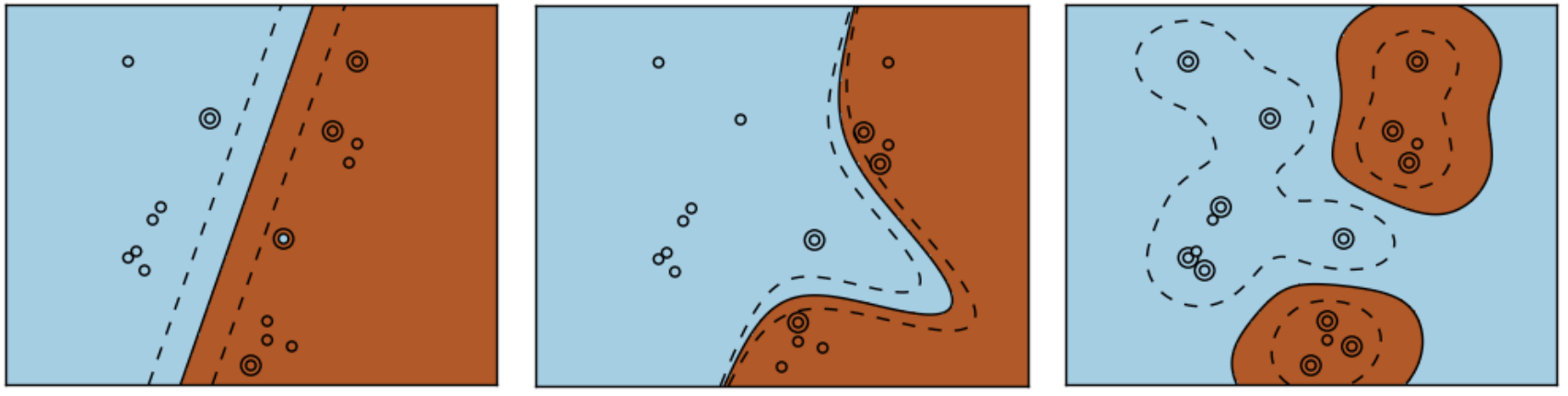
1. $K(x, x') = \big(\langle x, x' \rangle + 1\big)^d$
   — полиномиальная разделяющая поверхность степени $\leqslant d$;
2. $K(x, x') = \sigma\big(\langle x, x' \rangle\big)$
   — нейронная сеть с заданной функцией активации $\sigma(z)$
   (K не при всех σ является ядром);
3. $K(x, x') = \text{th}\big(k_1 \langle x, x' \rangle - k_0\big), \;\; k_0, k_1 \geqslant 0$
   — нейросеть с сигмоидными функциями активации;
4. $K(x, x') = \exp\big(-\gamma \|x - x'\|^2\big)$
   — сеть радиальных базисных функций (RBF ядро);

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)
$\langle x, x' \rangle$	$\left(\langle x, x' \rangle + 1\right)^d, \quad d{=}3$	$\exp\left(-\gamma \|x - x'\|^2\right)$



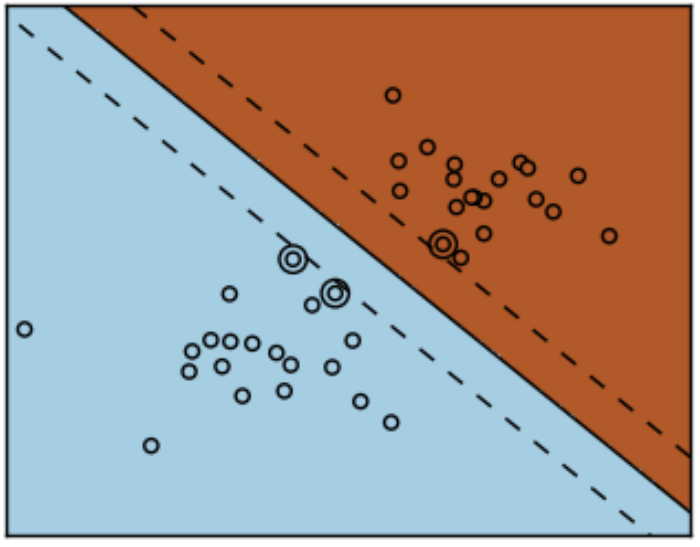
Пример из Python scikits learn: `http://scikit-learn.org/dev`

Влияние константы C на решение SVM

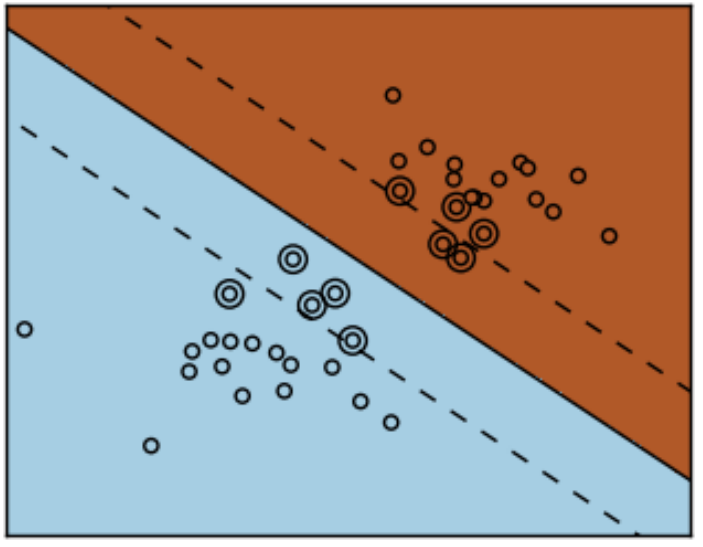
SVM — аппроксимация и регуляризация эмпирического риска:

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

большой C
слабая регуляризация



малый C
сильная регуляризация



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

Преимущества и недостатки SVM

Преимущества:

- Задача выпуклого квадратичного программирования имеет единственное решение.
- Выделяется множество опорных объектов.
- Имеются эффективные численные методы для SVM.
- Изящное обобщение на нелинейные классификаторы.

Недостатки:

- Опорными объектами могут становиться выбросы.
- Нет отбора признаков в исходном пространстве X.
- Приходится подбирать константу C.