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

Решив эту задачу численно относительно λ_i, получаем линейный классификатор:

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

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 $\lambda_i \neq 0$.

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

Решив эту задачу численно относительно λ_i, получаем линейный классификатор:

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

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 $\lambda_i \neq 0$.

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 $\psi\colon X \to H$ 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

$$a(x) = \mathrm{sign}\Bigl(\sum_{i=1}^{\ell} \lambda_i y_i K(x_i, x) - w_0 \Bigr).$$

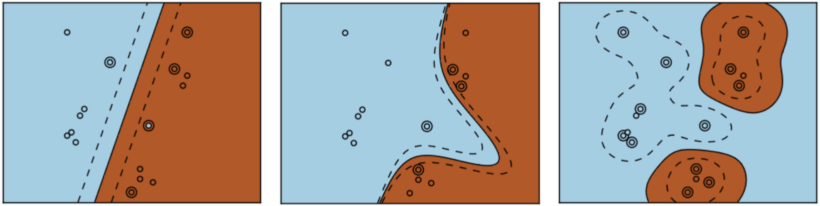
Ядра в SVM расширяют линейную модель классификации:

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

Гиперплоскость в спрямляющем пространстве соответствует нелинейной разделяющей поверхности в исходном.

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

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

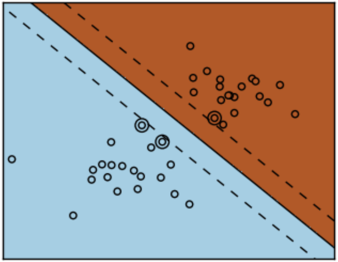


Пример из Python scikits learn: `http://scikit-learn.org/dev`

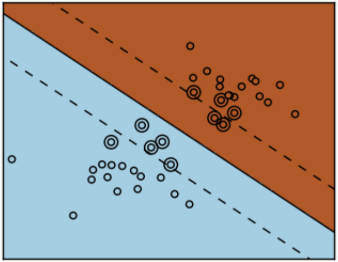
SVM — аппроксимация и регуляризация эмпирического риска:

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

большой C
слабая регуляризация



малый C
сильная регуляризация



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

Преимущества:

- 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.