Методы машинного обучения. Метод опорных векторов

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 • 22 октября 2024

Содержание

1. **Метод опорных векторов SVM**
   - Принцип оптимальной разделяющей гиперплоскости
   - Двойственная задача
   - Понятие опорного вектора
2. **Обобщения линейного SVM**
   - Ядра и спрямляющие пространства
   - SVM как двухслойная нейронная сеть
   - SVM-регрессия
3. **Регуляризация**
   - Подбор коэффициента регуляризации
   - Регуляризаторы для отбора признаков
   - Метод релевантных векторов RVM

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

$$a(x; w, w_0) = \operatorname{sign}\bigl(\langle x, w\rangle - w_0\bigr).$$

Критерий — минимизация эмпирического риска:

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

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

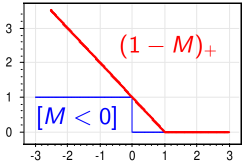
Аппроксимация и регуляризация эмпирического риска

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

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

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

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



Оптимальная разделяющая гиперплоскость

Линейный классификатор: $a(x, w) = \operatorname{sign}\bigl(\langle w, x\rangle - w_0\bigr)$

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

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

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

Разделяющая полоса (разделяющая гиперплоскость посередине):

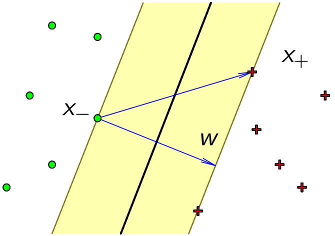
$$\bigl\{x\colon\ -1 \leqslant \langle w, x\rangle - w_0 \leqslant 1\bigr\}$$

$$\exists x_+: \quad \langle w, x_+\rangle - w_0 = +1$$

$$\exists x_-: \quad \langle w, x_-\rangle - w_0 = -1$$

Ширина полосы:

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



Обоснование кусочно-линейной функции потерь

Линейно разделимая выборка

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

Переход к линейно неразделимой выборке (эвристика)

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

Эквивалентная задача безусловной минимизации:

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

Напоминание. Условия Каруша–Куна–Таккера (KKT)

Задача математического программирования:

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

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

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

Применение условий ККТ к задаче SVM

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

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

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

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

Необходимые условия седловой точки функции Лагранжа

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

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

Необходимые условия седловой точки функции Лагранжа:

$$\frac{\partial \mathscr{L}}{\partial w} = w - \sum_{i=1}^{\ell} \lambda_i y_i x_i = 0 \quad \Longrightarrow \quad w = \sum_{i=1}^{\ell} \lambda_i y_i x_i;$$

$$\frac{\partial \mathscr{L}}{\partial w_0} = -\sum_{i=1}^{\ell} \lambda_i y_i = 0 \quad \Longrightarrow \quad \sum_{i=1}^{\ell} \lambda_i y_i = 0;$$

$$\frac{\partial \mathscr{L}}{\partial \xi_i} = -\lambda_i - \eta_i + C = 0 \quad \Longrightarrow \quad \eta_i + \lambda_i = C, \quad i = 1, \dots, \ell.$$

Понятие опорного вектора и типизация объектов

Система условий ККТ:

$$\begin{cases} w = \sum\limits_{i=1}^{\ell} \lambda_i y_i x_i; \quad \sum\limits_{i=1}^{\ell} \lambda_i y_i = 0; \quad M_i(w, w_0) \geqslant 1 - \xi_i; \\ \xi_i \geqslant 0, \quad \lambda_i \geqslant 0, \quad \eta_i \geqslant 0, \quad \eta_i + \lambda_i = C; \\ \lambda_i = 0 \;\text{ либо }\; M_i(w, w_0) = 1 - \xi_i; \\ \eta_i = 0 \;\text{ либо }\; \xi_i = 0; \end{cases}$$

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

Типизация объектов x_i, $i = 1, \ldots, \ell$:

1. $\lambda_i = 0$; $\eta_i = C$; $\xi_i = 0$; $M_i \geqslant 1$ — периферийный.
2. $0 < \lambda_i < C$; $0 < \eta_i < C$; $\xi_i = 0$; $M_i = 1$ — опорный-граничный
3. $\lambda_i = C$; $\eta_i = 0$; $\xi_i > 0$; $M_i < 1$ — опорный-нарушитель

Двойственная задача

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

Решение прямой задачи выражается через решение двойственной:

$$\begin{cases} w = \sum\limits_{i=1}^{\ell} \lambda_i y_i x_i; \\ w_0 = \langle w, x_i \rangle - y_i, \qquad \text{для любого } i\text{: } \lambda_i > 0,\ M_i = 1. \end{cases}$$

Линейный классификатор с признаками $f_i(x) = \langle x, x_i \rangle$:

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

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.

Двойственная задача. Нелинейное обобщение с ядром K

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

Решение прямой задачи выражается через решение двойственной:

$$\begin{cases} w = \sum\limits_{i=1}^{\ell} \lambda_i y_i x_i; \\ w_0 = \langle w, x_i \rangle - y_i, \qquad \text{для любого } i\text{: } \lambda_i > 0,\ M_i = 1. \end{cases}$$

Линейный классификатор с признаками $f_i(x) = K(x, x_i)$:

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

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.

Нелинейное обобщение SVM

Идея: заменить $\langle x, x' \rangle$ нелинейной функцией $K(x, x')$.

Переход к спрямляющему пространству,
как правило, более высокой размерности: $\psi\colon X \to H$.

Определение

Функция $K\colon X \times X \to \mathbb{R}$ — *ядро*, если $K(x, x') = \langle \psi(x), \psi(x') \rangle$ при некотором $\psi\colon X \to H$, где H — гильбертово пространство.

Теорема

Функция $K(x, x')$ является ядром тогда и только тогда, когда
- она симметрична: $K(x, x') = K(x', x)$;
- и неотрицательно определена:
  $\int_X \int_X K(x, x') g(x) g(x') dx dx' \geqslant 0$ для любой $g\colon X \to \mathbb{R}$.

Конструктивные методы синтеза ядер

1. $K(x, x') = \langle x, x' \rangle$ — ядро;
2. константа $K(x, x') = 1$ — ядро;
3. произведение ядер $K(x, x') = K_1(x, x')K_2(x, x')$ — ядро;
4. $\forall \psi : X \to \mathbb{R}$ произведение $K(x, x') = \psi(x)\psi(x')$ — ядро;
5. $K(x, x') = \alpha_1 K_1(x, x') + \alpha_2 K_2(x, x')$ при $\alpha_1, \alpha_2 > 0$ — ядро;
6. $\forall \varphi : X{\to}X$ если K_0 ядро, то $K(x, x') = K_0(\varphi(x), \varphi(x'))$ — ядро;
7. если $s \colon X \times X \to \mathbb{R}$ — симметричная интегрируемая функция, то $K(x, x') = \int_X s(x, z)s(x', z)\, dz$ — ядро;
8. если K_0 — ядро и функция $f \colon \mathbb{R} \to \mathbb{R}$ представима в виде сходящегося степенного ряда с неотрицательными коэффициентами, то $K(x, x') = f\bigl(K_0(x, x')\bigr)$ — ядро;

Пример: спрямляющее пространство для квадратичного ядра

Пусть $X = \mathbb{R}^2$, $K(u, v) = \langle u, v\rangle^2$, где $u = (u_1, u_2)$, $v = (v_1, v_2)$.

Задача: найти пространство H и преобразование $\psi\colon X \to H$, при которых $K(x, x') = \langle\psi(x), \psi(x')\rangle_H$.

Разложим квадрат скалярного произведения:

$$\begin{aligned} K(u, v) = \langle u, v\rangle^2 = \langle(u_1, u_2), (v_1, v_2)\rangle^2 &= \\ = (u_1 v_1 + u_2 v_2)^2 = u_1^2 v_1^2 + u_2^2 v_2^2 + 2u_1 v_1 u_2 v_2 &= \\ = \langle(u_1^2, u_2^2, \sqrt{2}u_1 u_2), (v_1^2, v_2^2, \sqrt{2}v_1 v_2)\rangle&. \end{aligned}$$

Таким образом,

$$H = \mathbb{R}^3, \quad \psi\colon (u_1, u_2) \mapsto \bigl(u_1^2, u_2^2, \sqrt{2}u_1 u_2\bigr),$$

Линейной поверхности в пространстве H соответствует квадратичная поверхность в исходном пространстве X.

Примеры ядер

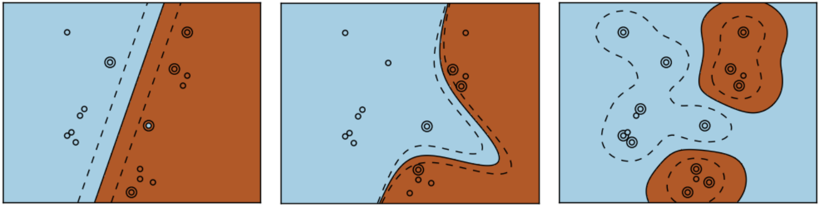
1. квадратичное ядро, $\dim H = \frac{1}{2} n(n+1)$
   $K(x, x') = \langle x, x' \rangle^2$
2. полиномиальное с мономами степени d, $\dim H = C^d_{n+d-1}$
   $K(x, x') = \langle x, x' \rangle^d$
3. полиномиальное с мономами степени $\leqslant d$
   $K(x, x') = \bigl(\langle x, x' \rangle + 1\bigr)^d$
4. нейросеть с сигмоидными функциями активации
   $K(x, x') = \mathrm{th}\bigl(k_1 \langle x, x' \rangle - k_0\bigr), \;\; k_0, k_1 \geqslant 0$
5. сеть радиальных базисных функций (RBF ядро)
   $K(x, x') = \exp\bigl(-\gamma \|x - x'\|^2\bigr)$

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

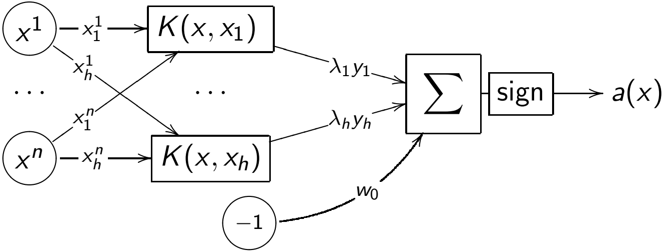


Пример из Python SkLearn: `http://scikit-learn.org/dev`

SVM: двухслойная нейросеть и метрический классификатор

Перенумеруем объекты так, чтобы $x_1, \dots, x_h$ были опорными.

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



Первый слой вместо скалярных произведений вычисляет ядра

Веса первого слоя — это сами опорные объекты

Метрический классификатор, если K — функция близости

Преимущества и недостатки SVM

Преимущества SVM перед двухслойными нейронными сетями:

- задача выпуклого квадратичного программирования имеет единственное решение
- число нейронов скрытого слоя определяется автоматически — это число опорных векторов

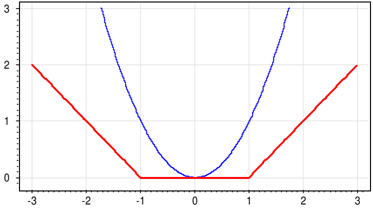
Недостатки классического SVM:

- нет общих подходов к оптимизации $K(x, x')$ под задачу
- на больших данных SVM обучается медленнее SG
- нет «встроенного» отбора признаков
- приходится подбирать константу C

SVM-регрессия

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

Функции потерь: $\mathscr{L}(\varepsilon) = \varepsilon^2,\ \ \mathscr{L}(\varepsilon) = \bigl(|\varepsilon| - \delta\bigr)_+,\ \ \varepsilon = a - y$



Постановка задачи:

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

Задача решается путём замены переменных
и сведения к задаче квадратичного программирования

SVM-регрессия

Замена переменных:

$$\xi_i^+ = \bigl(\langle w, x_i\rangle - w_0 - y_i - \delta\bigr)_+$$
$$\xi_i^- = \bigl(-\langle w, x_i\rangle + w_0 + y_i - \delta\bigr)_+$$

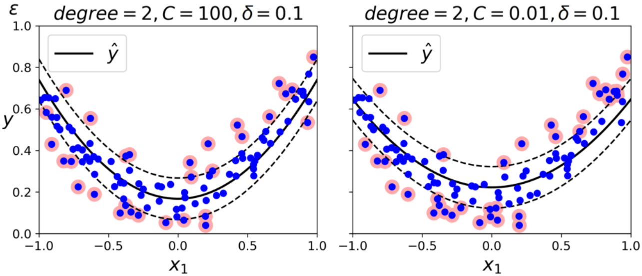
Постановка задачи SVM-регрессии:

$$\begin{cases} \frac{1}{2}\|w\|^2 + C\sum\limits_{i=1}^{\ell}(\xi_i^+ + \xi_i^-) \to \min\limits_{w, w_0, \xi^+, \xi^-} \\ y_i - \delta - \xi_i^- \leqslant \langle w, x_i\rangle - w_0 \leqslant y_i + \delta + \xi_i^+, \quad i = 1, \dots, \ell \\ \xi_i^- \geqslant 0, \quad \xi_i^+ \geqslant 0, \quad i = 1, \dots, \ell \end{cases}$$

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

SVM-регрессия. Пример 1

SVM-регрессия с полиномиальным ядром степени 2:

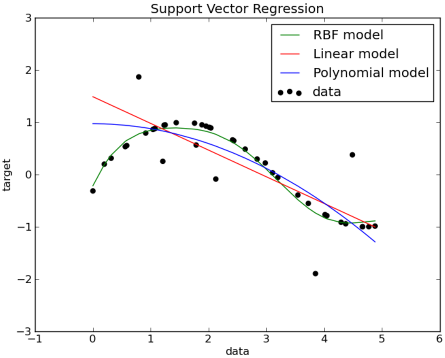


- Выделены опорные векторы
- Результат слабо зависит от константы C

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:



- Удачный выбор ядра имеет значение!

---

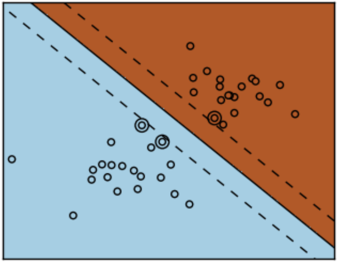
http://scikit-learn.org/0.5/auto_examples/svm/plot_svm_regression.html

Влияние константы C на решение SVM

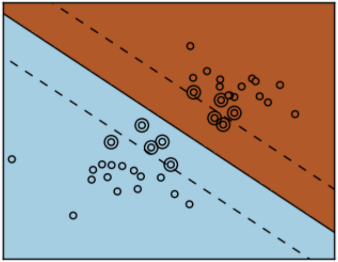
SVM — аппроксимация и регуляризация эмпирического риска:

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

большой C
слабая регуляризация



малый C
сильная регуляризация



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

Негладкие регуляризаторы для отбора и группировки признаков

Общий вид регуляризаторов (μ — параметр селективности):

$$\sum_{i=1}^{\ell}\bigl(1 - M_i(w, w_0)\bigr)_+ + \sum_{j=1}^{n} R_\mu(w_j) \;\to\; \min_{w}.$$

Регуляризаторы с эффектами отбора и группировки признаков:

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

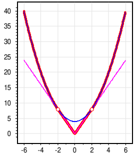
Elastic Net: $R_\mu(w) = \mu|w| + \tau w^2$

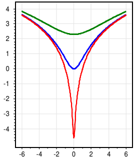
Support Feature Machine (SFM):

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

Relevance Feature Machine (RFM):

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





Метод релевантных векторов RVM (Relevance Vector Machine)

Положим, как и в SVM, при некоторых $\lambda_i \geqslant 0$

$$w = \sum_{i=1}^{\ell} \lambda_i y_i x_i,$$

причём опорным векторам x_i соответствуют $\lambda_i \neq 0$.

Проблема: Какие из коэффициентов λ_i лучше обнулить?

Идея: пусть регуляризатор зависит не от w, а от λ_i.
Пусть λ_i независимые, гауссовские, с дисперсиями α_i:

$$p(\lambda) = \frac{1}{(2\pi)^{\ell/2}\sqrt{\alpha_1 \cdots \alpha_\ell}} \exp\left(-\sum_{i=1}^{\ell} \frac{\lambda_i^2}{2\alpha_i}\right);$$

$$\sum_{i=1}^{\ell} \bigl(1 - M_i(w(\lambda), w_0)\bigr)_+ + \frac{1}{2}\sum_{i=1}^{\ell}\left(\ln \alpha_i + \frac{\lambda_i^2}{\alpha_i}\right) \to \min_{\lambda,\alpha}.$$

Преимущества и недостатки RVM

Преимущества:

- ⊕ Опорных векторов, как правило, меньше (более «разреженное» решение).
- ⊕ Шумовые выбросы уже не входят в число опорных.
- ⊕ Не надо искать параметр регуляризации (вместо этого α_i оптимизируются в процессе обучения).
- ⊕ Аналогично SVM, можно использовать ядра.

Недостатки:

- ⊖ Не всегда есть преимущество по качеству классификации.

M. E. Tipping. The relevance vector machine. 2000.

C. M. Bishop, M. E. Tipping. Variational relevance vector machine. 2000

Резюме по линейным классификаторам

- *SVM* — лучший метод линейной классификации
- С помощью *ядер* (kernel trick) SVM изящно обобщается для нелинейной классификации и нелинейной регрессии
- *Аппроксимация пороговой функции потерь* $\mathscr{L}(M)$ увеличивает зазор и повышает надёжность классификации
- *Регуляризация* увеличивает зазор, устраняет мультиколлинеарность и уменьшает переобучение
- *Негладкость функции потерь* приводит к отбору объектов
- *Негладкость регуляризатора* приводит к отбору признаков

В. Н. Вапник, А. Я. Лернер. Узнавание образов при помощи обобщенных портретов. 1963.

C. Cortes, V. Vapnik. Support vector networks. 1995.