

# Линейные методы классификации и регрессии: метод опорных векторов

К. В. Воронцов, А.В. Зухба  
vokov@forecsys.ru  
a\_\_l@mail.ru

март 2016

## 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

- Регуляризаторы для отбора признаков
- Вероятностная интерпретация регуляризации
- Метод релевантных векторов RVM

## Задача SVM — Support Vector Machine

Задача классификации:  $X = \mathbb{R}^n$ ,  $Y = \{-1, +1\}$ ,  
по обучающей выборке  $X^\ell = (x_i, y_i)_{i=1}^\ell$   
найти параметры  $w \in \mathbb{R}^n$ ,  $w_0 \in \mathbb{R}$  алгоритма классификации

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

**Метод минимизации эмпирического риска**  
с аппроксимацией пороговой функции потерь и регуляризацией:

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

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

Почему именно такая функция потерь? и такой регуляризатор?

## Оптимальная разделяющая гиперплоскость

Линейный классификатор:

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

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

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

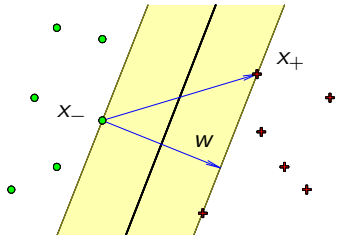
Нормировка:  $\min_{i=1, \dots, \ell} M_i(w, w_0) = 1.$

Разделяющая полоса:

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

Ширина полосы:

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



## Обоснование кусочно-линейной функции потерь

Линейно разделимая выборка

$$\begin{cases} \frac{1}{2} \|w\|^2 \rightarrow \min_{w, w_0}; \\ M_i(w, w_0) \geq 1, \quad i = 1, \dots, \ell. \end{cases}$$

Переход к линейно неразделимой выборке (эвристика)

$$\begin{cases} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{\ell} \xi_i \rightarrow \min_{w, w_0, \xi}; \\ M_i(w, w_0) \geq 1 - \xi_i, \quad i = 1, \dots, \ell; \\ \xi_i \geq 0, \quad i = 1, \dots, \ell. \end{cases}$$

Эквивалентная задача безусловной минимизации:

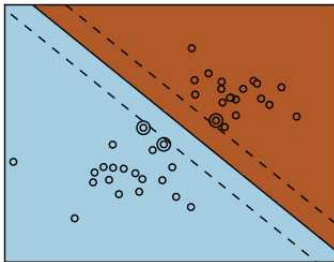
$$C \sum_{i=1}^{\ell} (1 - M_i(w, w_0))_+ + \frac{1}{2} \|w\|^2 \rightarrow \min_{w, w_0}.$$

## Влияние константы $C$ на решение SVM

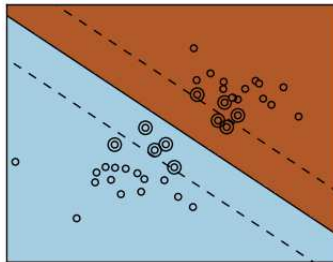
SVM — аппроксимация и регуляризация эмпирического риска:

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

большой  $C$   
слабая регуляризация



малый  $C$   
сильная регуляризация



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

## Напоминание. Условия Каруша–Куна–Таккера

Задача математического программирования:

$$\begin{cases} f(x) \rightarrow \min_x; \\ g_i(x) \leq 0, & i = 1, \dots, m; \\ h_j(x) = 0, & j = 1, \dots, k. \end{cases}$$

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

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial x} = 0, & \mathcal{L}(x; \mu, \lambda) = f(x) + \sum_{i=1}^m \mu_i g_i(x) + \sum_{j=1}^k \lambda_j h_j(x); \\ g_i(x) \leq 0; & h_j(x) = 0; \text{ (исходные ограничения)} \\ \mu_i \geq 0; & \text{ (двойственные ограничения)} \\ \mu_i g_i(x) = 0; & \text{ (условие дополняющей нежёсткости)} \end{cases}$$

## Применение условий ККТ к задаче SVM

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

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

$\lambda_i$  — переменные, двойственные к ограничениям  $M_i \geq 1 - \xi_i$ ;  
 $\eta_i$  — переменные, двойственные к ограничениям  $\xi_i \geq 0$ .

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



## Необходимые условия седловой точки функции Лагранжа

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

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

Необходимые условия седловой точки функции Лагранжа:

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

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

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

## Понятие опорного вектора

Типизация объектов:

1.  $\lambda_i = 0$ ;  $\eta_i = C$ ;  $\xi_i = 0$ ;  $M_i \geq 1$ .  
— периферийные (неинформативные) объекты.
2.  $0 < \lambda_i < C$ ;  $0 < \eta_i < C$ ;  $\xi_i = 0$ ;  $M_i = 1$ .  
— **опорные** граничные объекты.
3.  $\lambda_i = C$ ;  $\eta_i = 0$ ;  $\xi_i > 0$ ;  $M_i < 1$ .  
— **опорные**-нарушители.

### Определение

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

## Двойственная задача

$$\begin{cases} -\mathcal{L}(\lambda) = -\sum_{i=1}^{\ell} \lambda_i + \frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \lambda_i \lambda_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \rightarrow \min_{\lambda}; \\ 0 \leq \lambda_i \leq C, \quad i = 1, \dots, \ell; \\ \sum_{i=1}^{\ell} \lambda_i y_i = 0. \end{cases}$$

Решение прямой задачи выражается через решение двойственной:

$$\begin{cases} \mathbf{w} = \sum_{i=1}^{\ell} \lambda_i y_i \mathbf{x}_i; \\ w_0 = \langle \mathbf{w}, \mathbf{x}_i \rangle - y_i, \quad \text{для любого } i: \lambda_i > 0, \quad M_i = 1. \end{cases}$$

Линейный классификатор:

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

## Нелинейное обобщение SVM

Переход к спрямляющему пространству  
более высокой размерности:  $\psi: X \rightarrow H$ .

### Определение

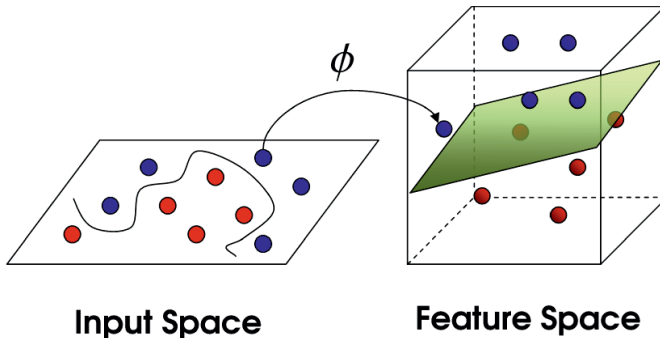
Функция  $K: X \times X \rightarrow \mathbb{R}$  — *ядро*, если  $K(x, x') = \langle \psi(x), \psi(x') \rangle$  при некотором  $\psi: X \rightarrow H$ , где  $H$  — гильбертово пространство.

### Теорема

Функция  $K(x, x')$  является ядром тогда и только тогда, когда она симметрична:  $K(x, x') = K(x', x)$ ;  
и неотрицательно определена:

$$\int_X \int_X K(x, x') g(x) g(x') dx dx' \geq 0 \text{ для любой } g: X \rightarrow \mathbb{R}.$$

## Переход к спрямляющему пространству



## Конструктивные методы синтеза ядер

- ❶  $K(x, x') = \langle x, x' \rangle$  — ядро;
- ❷ константа  $K(x, x') = 1$  — ядро;
- ❸ произведение ядер  $K(x, x') = K_1(x, x')K_2(x, x')$  — ядро;
- ❹  $\forall \psi : X \rightarrow \mathbb{R}$  произведение  $K(x, x') = \psi(x)\psi(x')$  — ядро;
- ❺  $K(x, x') = \alpha_1 K_1(x, x') + \alpha_2 K_2(x, x')$  при  $\alpha_1, \alpha_2 > 0$  — ядро;
- ❻  $\forall \varphi : X \rightarrow X$  если  $K_0$  ядро, то  $K(x, x') = K_0(\varphi(x), \varphi(x'))$  — ядро;
- ❼ если  $s : X \times X \rightarrow \mathbb{R}$  — симметричная интегрируемая функция, то  $K(x, x') = \int_X s(x, z)s(x', z) dz$  — ядро;
- ❽ если  $K_0$  — ядро и функция  $f : \mathbb{R} \rightarrow \mathbb{R}$  представима в виде сходящегося степенного ряда с неотрицательными коэффициентами, то  $K(x, x') = f(K_0(x, x'))$  — 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  $\psi: X \rightarrow H$ , 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,

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

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

## Примеры ядер

- ❶  $K(x, x') = \langle x, x' \rangle^2$   
— квадратичное ядро;
- ❷  $K(x, x') = \langle x, x' \rangle^d$   
— полиномиальное ядро с мономами степени  $d$ ;
- ❸  $K(x, x') = (\langle x, x' \rangle + 1)^d$   
— полиномиальное ядро с мономами степени  $\leq d$ ;
- ❹  $K(x, x') = \sigma(\langle x, x' \rangle)$   
— нейросеть с заданной функцией активации  $\sigma(z)$   
(не при всех  $\sigma$  является ядром);
- ❺  $K(x, x') = \text{th}(k_1 \langle x, x' \rangle - k_0), \quad k_0, k_1 \geq 0$   
— нейросеть с сигмоидными функциями активации;
- ❻  $K(x, x') = \exp(-\beta \|x - x'\|^2)$   
— сеть радиальных базисных функций (RBF ядро);

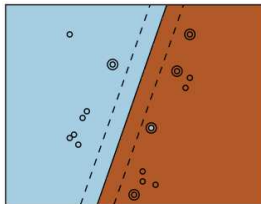


## Классификация с различными ядрами

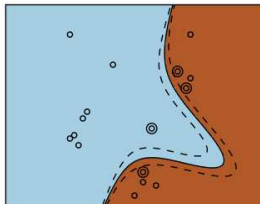
Гиперплоскость в спрямляющем пространстве соответствует нелинейной разделяющей поверхности в исходном.

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

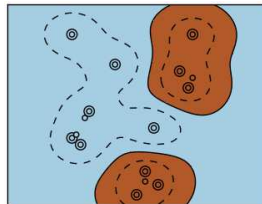
линейное  
 $\langle x, x' \rangle$



полиномиальное  
 $(\langle x, x' \rangle + 1)^d, d=3$



гауссовское (RBF)  
 $\exp(-\beta \|x - x'\|^2)$

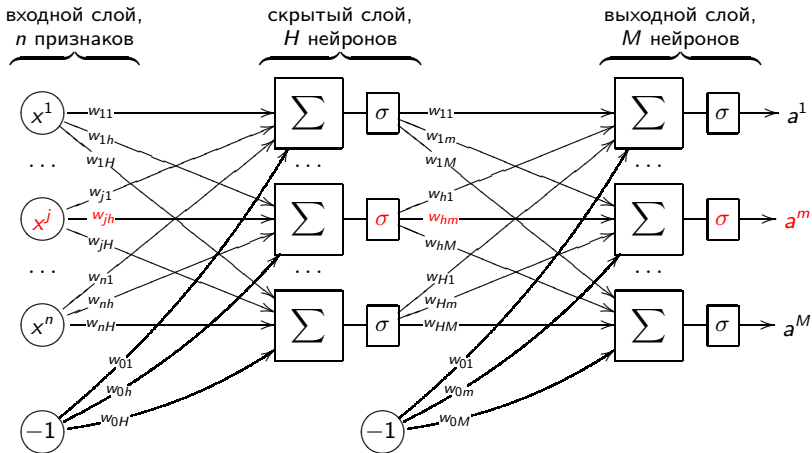


---

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

## Двухслойная нейронная сеть

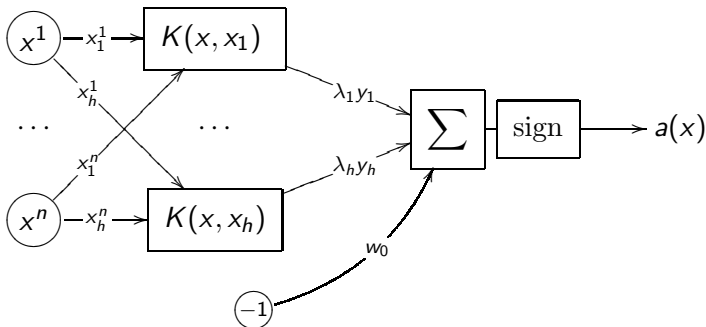
$$a^m(x) = \sigma \left( \sum_{h=0}^H w_{hm} \sigma \left( \sum_{j=0}^J w_{jh} f_j(x) \right) \right), \quad \sigma - \text{функция активации}$$



## SVM как двухслойная нейронная сеть

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

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



Первый слой вместо скалярных произведений вычисляет ядра

## Преимущества и недостатки SVM

Преимущества SVM перед SG и нейронными сетями:

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

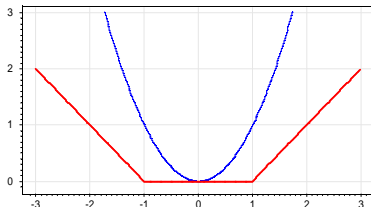
Недостатки классического SVM:

- Нет общих подходов к оптимизации  $K(x, x')$  под задачу.
- Нет «встроенного» отбора признаков.
- Приходится подбирать константу  $C$ .

## SVM-регрессия

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

Функция потерь:  $\mathcal{L}(\varepsilon) = (|\varepsilon| - \delta)_+$  в сравнении с  $\mathcal{L}(\varepsilon) = \varepsilon^2$ :



Постановка задачи:

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

Задача решается путём замены переменных  
и сведения к задаче квадратичного программирования

## SVM-регрессия

Замена переменных:

$$\begin{aligned}\xi_i^+ &= (\langle w, x_i \rangle - w_0 - y_i - \delta)_+; \\ \xi_i^- &= (-\langle w, x_i \rangle + w_0 + y_i - \delta)_+;\end{aligned}$$

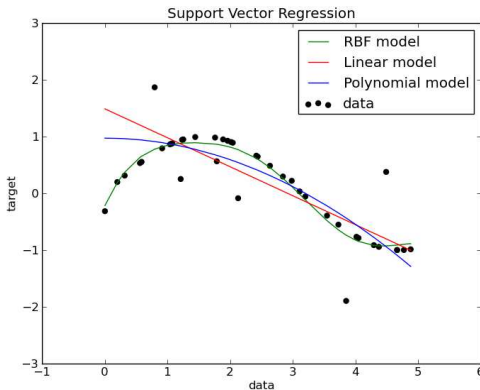
Постановка задачи SVM-регрессии:

$$\begin{cases} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{\ell} (\xi_i^+ + \xi_i^-) \rightarrow \min_{w, w_0, \xi^+, \xi^-}; \\ y_i - \delta - \xi_i^- \leq \langle w, x_i \rangle - w_0 \leq y_i + \delta + \xi_i^+, \quad i = 1, \dots, \ell; \\ \xi_i^- \geq 0, \quad \xi_i^+ \geq 0, \quad i = 1, \dots, \ell. \end{cases}$$

Это задача квадратичного программирования с линейными ограничениями-неравенствами, решается также сведением к двойственной задаче.

## Пример из Python scikits learn

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



[http://scikit-learn.org/0.5/auto\\_examples/svm/plot\\_svm\\_regression.html](http://scikit-learn.org/0.5/auto_examples/svm/plot_svm_regression.html)

## 1-norm SVM (LASSO SVM)

Аппроксимация эмпирического риска с  $L_1$ -регуляризацией:

$$\sum_{i=1}^{\ell} (1 - M_i(w, w_0))_+ + \mu \sum_{j=1}^n |w_j| \rightarrow \min_{w, w_0}.$$

- ⊕ Отбор признаков с параметром *селективности*  $\mu$ :  
чем больше  $\mu$ , тем меньше признаков останется
- ⊖ LASSO начинает отбрасывать значимые признаки,  
когда ещё не все шумовые отброшены
- ⊖ Нет *эффекта группировки* (grouping effect):  
значимые зависимые признаки должны отбираться вместе  
и иметь примерно равные веса  $w_j$

---

Bradley P., Mangasarian O. Feature selection via concave minimization and support vector machines // ICML 1998.



## 1-norm SVM (LASSO SVM)

Аппроксимация эмпирического риска с  $L_1$ -регуляризацией:

$$\sum_{i=1}^{\ell} (1 - M_i(w, w_0))_+ + \mu \sum_{j=1}^n |w_j| \rightarrow \min_{w, w_0}.$$

Почему  $L_1$ -регуляризатор приводит к отбору признаков?

Замена переменных:  $u_j = \frac{1}{2}(|w_j| + w_j)$ ,  $v_j = \frac{1}{2}(|w_j| - w_j)$ .

Тогда  $w_j = u_j - v_j$  и  $|w_j| = u_j + v_j$ ;

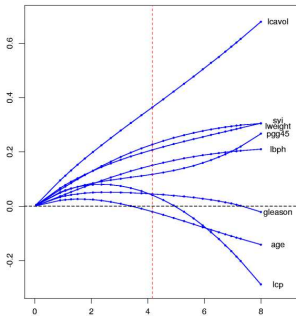
$$\begin{cases} \sum_{i=1}^{\ell} (1 - M_i(u - v, w_0))_+ + \mu \sum_{j=1}^n (u_j + v_j) \rightarrow \min_{u, v} \\ u_j \geq 0, \quad v_j \geq 0, \quad j = 1, \dots, n; \end{cases}$$

чем больше  $\mu$ , тем больше индексов  $j$  таких, что  $u_j = v_j = 0$ , но тогда  $w_j = 0$ , значит, **признак не учитывается**.

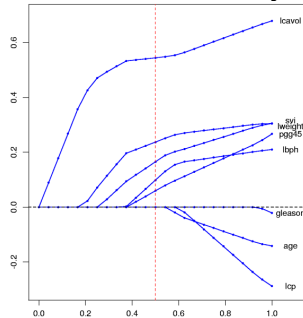
## Сравнение $L_2$ и $L_1$ регуляризации

Зависимость весов  $w_j$  от коэффициента  $\frac{1}{\mu}$

$L_2$  регуляризатор:  $\mu \sum_j w_j^2$



$L_1$  регуляризатор:  $\mu \sum_j |w_j|$



Задача из UCI: prostate cancer (диагностика рака)

*T.Hastie, R.Tibshirani, J.Friedman. The Elements of Statistical Learning. Springer, 2001.*

## Doubly Regularized SVM (Elastic Net SVM)

$$C \sum_{i=1}^{\ell} (1 - M_i(w, w_0))_+ + \mu \sum_{j=1}^n |w_j| + \frac{1}{2} \sum_{j=1}^n w_j^2 \rightarrow \min_{w, w_0}.$$

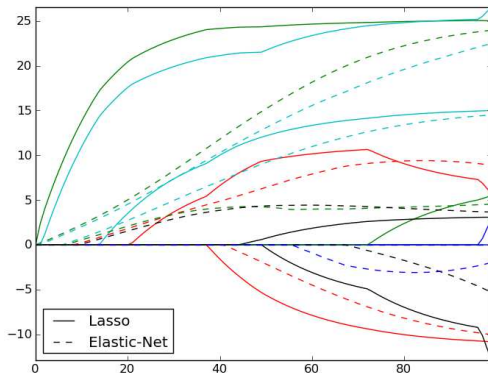
- ⊕ Отбор признаков с параметром *селективности*  $\mu$ :  
чем больше  $\mu$ , тем меньше признаков останется
- ⊕ Есть эффект группировки
- ⊖ Шумовые признаки также группируются вместе,  
и группы значимых признаков могут отбрасываться,  
когда ещё не все шумовые отброшены

---

*Li Wang, Ji Zhu, Hui Zou.* The doubly regularized support vector machine // *Statistica Sinica*, 2006. N16, Pp. 589–615.

## Doubly Regularized SVM (Elastic Net SVM)

Elastic Net менее жёстко отбирает признаки.  
Зависимости весов  $w_j$  от коэффициента  $\log \frac{1}{\mu}$ :



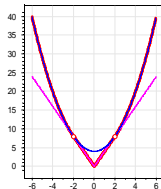
Пример из Python scikits learn:

[scikit-learn.org/0.5/auto\\_examples/glm/plot\\_lasso\\_coordinate\\_descent\\_path.html](http://scikit-learn.org/0.5/auto_examples/glm/plot_lasso_coordinate_descent_path.html)

## Support Features Machine (SFM)

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

$$R_{\mu}(w_j) = \begin{cases} 2\mu|w_j|, & |w_j| \leq \mu; \\ \mu^2 + w_j^2, & |w_j| \geq \mu; \end{cases}$$



- ⊕ Отбор признаков с параметром *селективности*  $\mu$
- ⊕ Есть эффект группировки
- ⊕ Значимые зависимые признаки ( $|w_j| > \mu$ ) группируются и входят в решение совместно (как в Elastic Net),
- ⊕ Шумовые признаки ( $|w_j| < \mu$ ) подавляются независимо (как в LASSO)

---

*Tatarchuk A., Urlov E., Mottl V., Windridge D. A support kernel machine for supervised selective combining of diverse pattern-recognition modalities // Multiple Classifier Systems. LNCS, Springer-Verlag, 2010. Pp.165–174.*

## Relevance Features Machine (RFM)

$$C \sum_{i=1}^{\ell} (1 - M_i(w, w_0))_+ + \sum_{j=1}^n \ln(w_j^2 + \frac{1}{\mu}) \rightarrow \min_{w, w_0}.$$

- ⊕ Отбор признаков с параметром *селективности*  $\mu$ :  
чем больше  $\mu$ , тем меньше признаков останется
- ⊕ Есть эффект группировки
- ⊕ Лучше отбирает набор значимых признаков, когда  
они только совместно обеспечивают хорошее решение

---

*Tatarchuk A., Mottl V., Eliseyev A., Windridge D.* Selectivity supervision in combining pattern recognition modalities by feature- and kernel-selective Support Vector Machines // 19th International Conference on Pattern Recognition, Vol 1-6, 2008, Pp. 2336–2339.

## Принцип максимума правдоподобия

Пусть  $X \times Y$  — в.п. с плотностью  $p(x, y|w)$ .

Пусть  $X^\ell$  — простая (i.i.d.) выборка:  $(x_i, y_i)_{i=1}^\ell \sim p(x, y|w)$

- Максимизация правдоподобия:

$$L(w; X^\ell) = \ln \prod_{i=1}^{\ell} p(x_i, y_i|w) = \sum_{i=1}^{\ell} \ln p(x_i, y_i|w) \rightarrow \max_w.$$

- Минимизация аппроксимированного эмпирического риска:

$$Q(w; X^\ell) = \sum_{i=1}^{\ell} \mathcal{L}(y_i g(x_i, w)) \rightarrow \min_w;$$

- Эти два принципа эквивалентны, если положить

$$-\ln p(x_i, y_i|w) = \mathcal{L}(y_i g(x_i, w)).$$

$$\boxed{\text{модель } p} \Leftrightarrow \boxed{\text{модель } g \text{ и функция потерь } \mathcal{L}}.$$

## Обобщение: байесовская регуляризация

$p(x, y|w)$  — вероятностная модель данных;

$p(w; \gamma)$  — априорное распределение параметров модели;

$\gamma$  — вектор *гиперпараметров*;

Теперь не только появление выборки  $X^\ell$ ,  
но и появление модели  $w$  также полагается случайным.

Совместное правдоподобие данных и модели:

$$p(X^\ell, w) = p(X^\ell|w) p(w; \gamma).$$

*Принцип максимума совместного правдоподобия:*

$$L(w, X^\ell) = \ln p(X^\ell, w) = \sum_{i=1}^{\ell} \ln p(x_i, y_i|w) + \underbrace{\ln p(w; \gamma)}_{\text{регуляризатор}} \rightarrow \max_{w, \gamma}.$$



## Пример 1: квадратичный (гауссовский) регуляризатор

Пусть  $w \in \mathbb{R}^n$  имеет  $n$ -мерное гауссовское распределение:

$$p(w; \sigma) = \frac{1}{(2\pi\sigma)^{n/2}} \exp\left(-\frac{\|w\|^2}{2\sigma}\right), \quad \|w\|^2 = \sum_{j=1}^n w_j^2,$$

т. е. все веса независимы, имеют нулевое матожидание и равные дисперсии  $\sigma$ ;  $\sigma$  — гиперпараметр.

Логарифмируя, получаем квадратичный регуляризатор:

$$-\ln p(w; \sigma) = \frac{1}{2\sigma} \|w\|^2 + \text{const}(w).$$

Вероятностный смысл параметра регуляризации:  $C = \sigma$ .

## Пример 2: лапласовский регуляризатор

Пусть  $w \in \mathbb{R}^n$  имеет  $n$ -мерное распределение Лапласа:

$$p(w; C) = \frac{1}{(2C)^n} \exp\left(-\frac{\|w\|_1}{C}\right), \quad \|w\|_1 = \sum_{j=1}^n |w_j|,$$

т. е. все веса независимы, имеют нулевое матожидание и равные дисперсии;  $C$  — гиперпараметр.

Логарифмируя, получаем регуляризатор по  $L_1$ -норме:

$$-\ln p(w; C) = \frac{1}{C} \sum_{j=1}^n |w_j| + \text{const}(w).$$

Вероятностный смысл параметра регуляризации:  $\mu = \frac{1}{\sigma}$ .

## Метод релевантных векторов RVM (Relevance Vector Machine)

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

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

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

**Проблема:** Какие из коэффициентов  $\lambda_i$  лучше обнулить?

**Идея:** пусть регуляризатор зависит не от  $w$ , а от  $\lambda_i$ .

Пусть  $\lambda_i$  независимые, гауссовские, с дисперсиями  $\alpha_i$ :

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

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

## 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  $\alpha$  оптимизируются в процессе обучения).
- ⊕ Аналогично SVM, можно использовать ядра.

### Недостатки:

- ⊖ Авторам не удалось показать практическое преимущество по качеству классификации.

## Резюме по линейным классификаторам

- *SVM* — лучший метод линейной классификации
- *SVM* изящно обобщается для нелинейной классификации, для линейной и нелинейной регрессии
- *Аппроксимация пороговой функции потерь  $\mathcal{L}(M)$*  увеличивает зазор и повышает качество классификации
- *Регуляризация* устраняет мультиколлинеарность и переобучение
- *Регуляризация* эквивалентна введению априорного распределения в пространстве коэффициентов
- $L_1$  и другие нестандартные регуляризаторы делают отбор признаков без явного перебора подмножеств