Support Vector Machines

Polina Polunina

Based on the lectures of K. Vorontsov
http://shad.yandex.ru/lectures

SVM Classification Problem

- Dataset $X^l = (x_i, y_i)_{i=1}^l$
- x_i - objects, vectors from the set $X = \mathbb{R}^n$
- y_i - class labels, elements of the set $Y = \{-1, +1\}$
- Problem statement: given the dataset, find parameters $w \in \mathbb{R}^n, w_0 \in \mathbb{R}$ such that:
$$a(x; w, w_0) = sign(\langle x, w\rangle - w_0)$$
- Criterion: empirical risk minimization:
$$\sum_{i=1}^{l} [a(x_i; w, w_0) \neq y_i] = \sum_{i=1}^{l} [M_i(w, w_0) < 0] \to \min_{w, w_0},$$
Where $M_i(w, w_0) = (\langle x_i, w\rangle - w_0) y_i$ - margin of x_i,

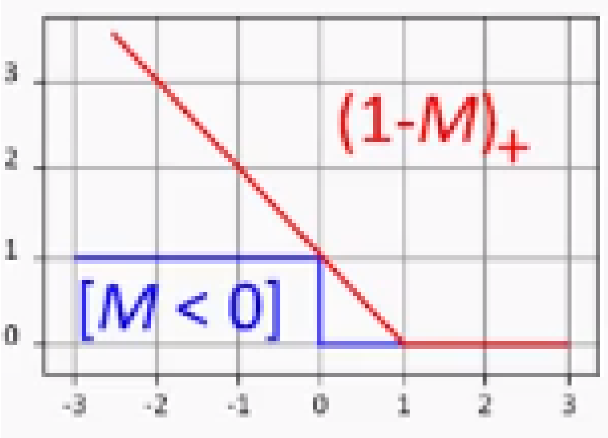
$b(x) = \langle x_i, w\rangle - w_0$ - discriminant function

SVM Classification Problem: Approximation and Empirical Risk Minimization

- Empirical Risk is a piecewise constant function
- Change it by its upper estimate, that is continuous by parameters:

$$Q(w, w_0) = \sum_{i=1}^{l} [M_i(w, w_0) < 0] \leq \sum_{i=1}^{l} [1 - M_i(w, w_0)]_+$$

- *Approximation* penalizes objects approaching the class boundary and increases the gap between the classes
- *Regularization* penalizes unstable solutions in case of multicollinearity



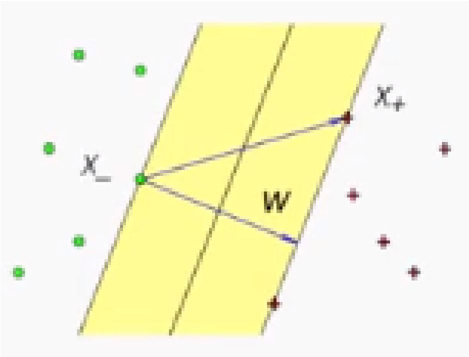
SVM Classification Problem: Optimal Separation Hyperplane

- Linear classifier:

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

- Suppose that the dataset $X^l = (x_i, y_i)_{i=1}^l$ is linearly separable:

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



- Normalization: $\min_{i=1,\dots,l} M_i(w, w_0) = 1$

- Separating strip:

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

- Width of the strip:

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

SVM Classification Problem: NOT Linearly Separable Dataset

- Problem statement in a linearly separable case:

$$\begin{cases} \frac{1}{2}\|w\|^2 \to \min\limits_{w,w_0}; \\ M_i(w,w_0) \geq 1, \qquad i = 1, \dots, l \end{cases}$$

- Generalization – NOT linearly separable case:

$$\begin{cases} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} \xi_i \to \min\limits_{w,w_0,\xi}; \\ M_i(w,w_0) \geq 1 - \xi_i, \qquad i = 1, \dots, l; \\ \xi_i \geq 0, i = 1, \dots, l \end{cases}$$

- Excluding ξ_i, we get the unconstrained optimization problem:

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

Remainder: Karush–Kuhn–Tucker Conditions

- Mathematical programming problem:

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

- Necessary conditions. If x^* – a local minimum point,

then there exists multipliers $\mu_i, i = 1, \dots, m;\ \lambda_j, j = 1, \dots, k$ such that:

Lagrangian: $L(x; \mu, \lambda) = f(x) + \sum_{i=1}^{m} \mu_i g_i(x) + \sum_{j=1}^{k} \lambda_j h_j(x)$

First Order Conditions:

$$\begin{cases} \dfrac{\partial L}{\partial x^*} = 0 \\ g_i(x^*) \le 0, \qquad i = 1, \dots, m \\ h_j(x^*) = 0, \qquad j = 1, \dots, k \\ \mu_i \ge 0, \qquad i = 1, \dots, m \\ \mu_i g_i(x^*) = 0, \qquad i = 1, \dots, m \end{cases}$$

- primal feasibility

- dual feasibility

- complementary slackness

KKT Conditions and SVM Problem

- Lagrangian: $L(w, w_0, \xi; \lambda, \eta) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{l} \lambda_i (M_i(w, w_0) - 1) - \sum_{i=1}^{l} \xi_i (\lambda_i + \eta_i - C)$,

  λ_i - variables, that are dual to the constraints $M_i \geq 1 - \xi_i$;

  η_i - variables, that are dual to the constraints $\xi_i \geq 0$.

- First Order Conditions:

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

Dual Problem

- $$\begin{cases} -\sum_{i=1}^{l} \lambda_i + \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \lambda_i\lambda_j y_i y_j \langle x_i, x_j \rangle \to \min_\lambda \ ; \\ 0 \le \lambda_i \le C, \quad i = 1, \dots, l; \\ \sum_{i=1}^{l} \lambda_i \, y_i = 0. \end{cases}$$
- Solving this problem numerically with respect to λ_i, we get the linear classifier:

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

where $w_0 = \sum_{i=1}^{l} \lambda_i y_i \langle x_i, x_j \rangle - y_j$ for such j that $\lambda_j > 0, M_j = 1$

- Definition:

Object x_i is called *supporting* if $\lambda_i \neq 0$.

SVM: Advantages and Disadvantages

Advantages:

- Convex quadratic programming problem has a unique solution
- A set of supporting objects exists
- There are effective numerical methods for SVM
- Generalization for non linear classifiers

Disadvantages:

- Outliers could be chosen as supporting objects
- There is no feature selection in X
- It is necessary to select a value for the constant C

SVM classification: non linear generalization

- $$\begin{cases} -\sum_{i=1}^{l}\lambda_i + \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\lambda_i\lambda_j y_i y_j K(x_i, x_j) \to \min_\lambda \ ; \\ 0 \le \lambda_i \le C, \quad i = 1, \dots, l; \\ \sum_{i=1}^{l}\lambda_i\, y_i = 0. \end{cases}$$

- Solving this problem numerically with respect to λ_i, we get the linear classifier:

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

where $w_0 = \sum_{i=1}^{l}\lambda_i y_i K(x_i, x_j) - y_j$ for such j that $\lambda_j > 0, M_j = 1$

Kernels for Non-linear SVM Generalization

- Definition:

A function $K(x, x')$ is called a *kernel*, if it could be represented as a scalar multiplication:

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

for some mapping $\varphi: X \to H$ from a feature space X to a new rectifying space H (Hilbert space).

- Possible interpretation:

a feature $f_i(x) = K(x_i, x)$ is an estimate of how close an object x is located to a supporting object x_i. Choosing supporting objects, SVM performs feature selection in a linear classifier

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

- Theorem:

A function $K(x, x')$ is a kernel if and only if it is symmetrical: $K(x, x') = K(x', x)$

and non-negative: $\int_X \int_X K(x, x') g(x) g(x')\, dx dx' \geq 0 \quad \forall\, g : X \to \mathbb{R}$.

Kernels Examples

Kernels (in SVM) expand the linear classification model:

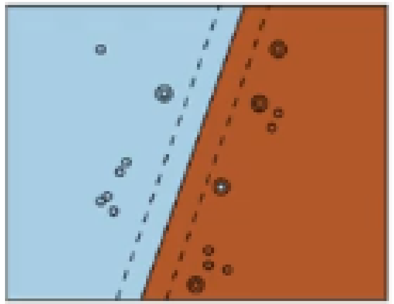
- $K(x,x') = (\langle x,x'\rangle + 1)^d$ - polynomial separating surface of degree $\leq d$
- $K(x,x') = \sigma(\langle x,x'\rangle)$ - a neural network with activation function $\sigma(z)$.
  (There are some σ such that K is not a kernel)
- $K(x,x') = th(k_1\langle x,x'\rangle - k_0)$, $k_0,\, k_1 \geq 0$ - a neural network with sigmoidal activation functions
- $K(x,x') = exp(-\gamma\|x - x'\|^2)$ - radial basis functions (RBF Kernel)

Classification with Different Kernels

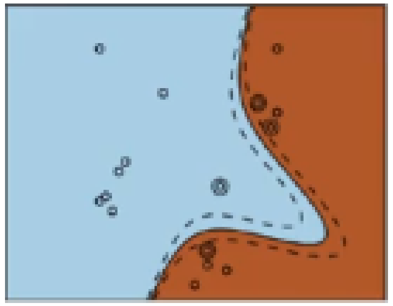
- Hyperplane in a rectifying space is equal to a non-linear separating surface in the original space.

- Some examples with different kernels $K(x, x')$

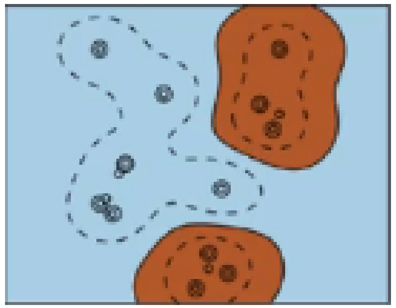
linear
$\langle x, x' \rangle$



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



Gaussian (RBF)
$exp(-\gamma \|x - x'\|^2)$

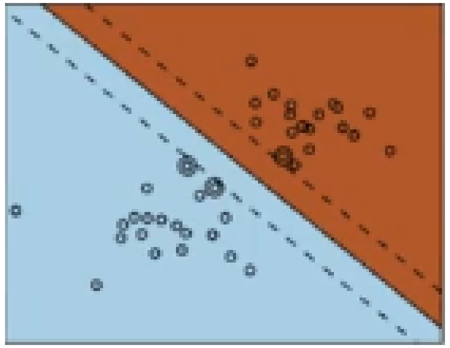


Influence of the Constant C on SVM Solutions

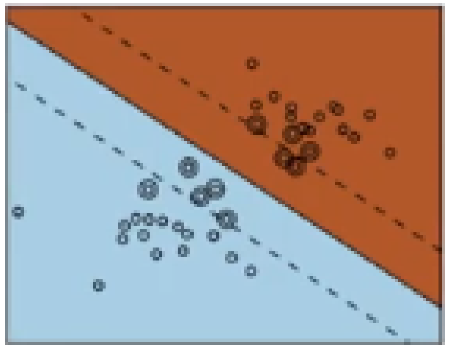
- SVM is an approximation and regularization of the empirical risk:

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

large C
weak regularization



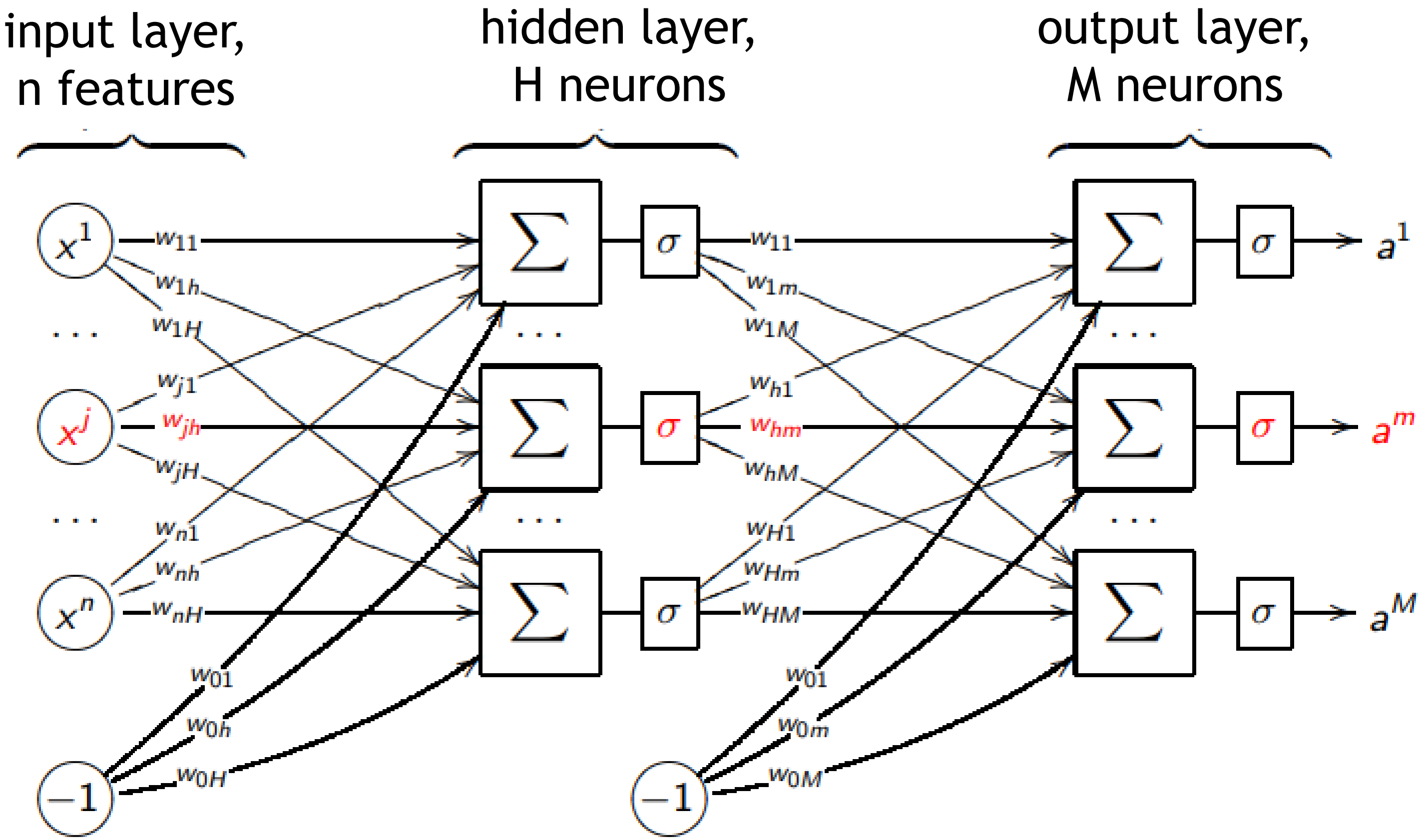
small C
strong regularization



SVM as a Neural Network

- Neural Network with two layers:

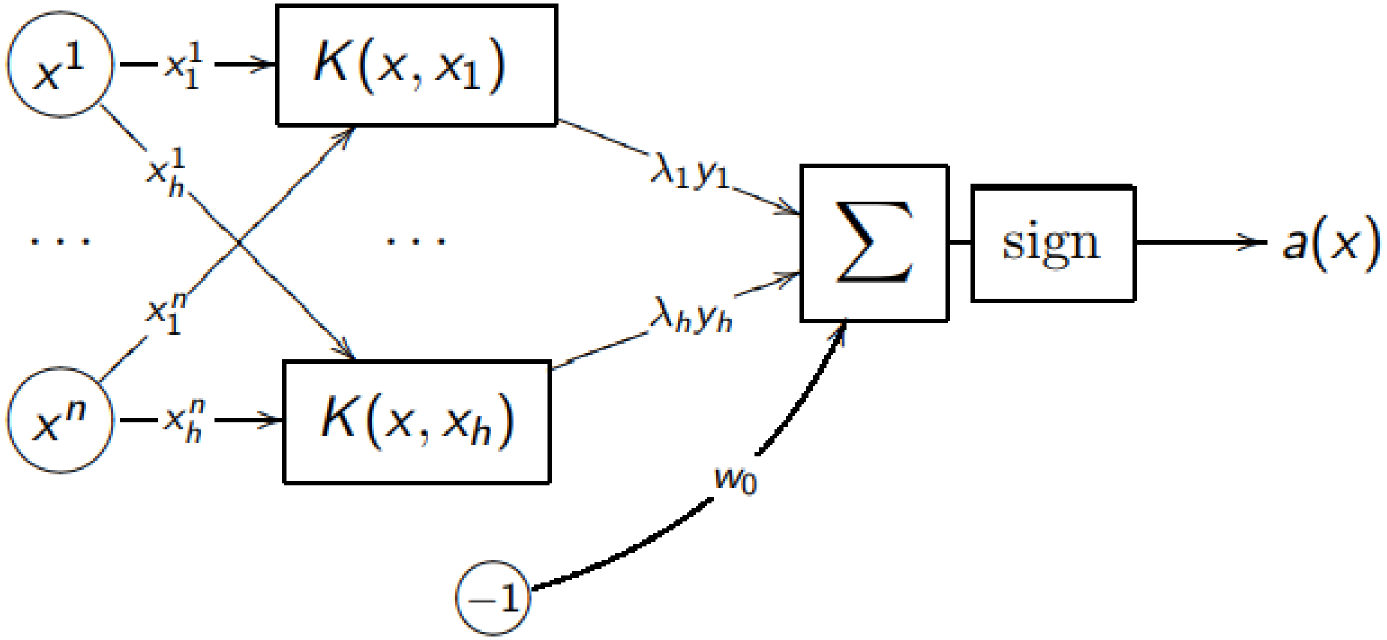
$a^m(x) = \sigma(\sum_{h=0}^{H} w_{hm}\sigma\left(\sum_{j=0}^{J} w_{jh} f_i(x)\right)$, σ - activation function



SVM as a Neural Network

- Re-number the objects such that $x_1, \dots, x_h$ become supporting objects:

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



The first layer calculates kernels instead of scalar multiplications

SVM: Advantages and Disadvantages

Advantages:

- Convex quadratic programming problem has a unique solution
- The number of hidden layer neurons is defined automatically as the number of supporting vectors

Disadvantages:

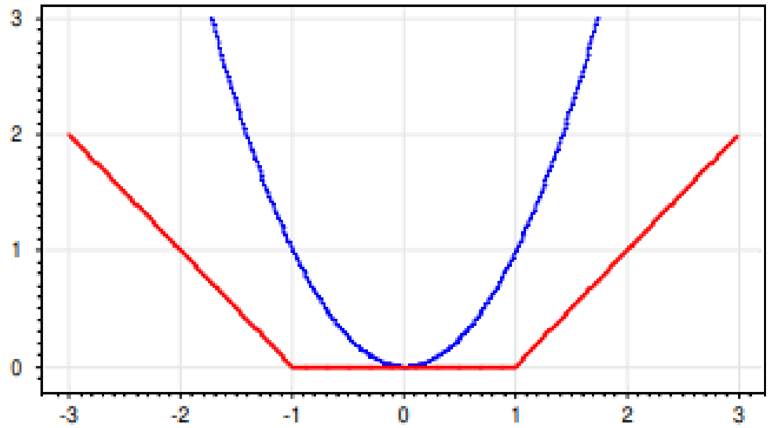
- There is no common practice to kernel optimization from task to task
- There is no feature selection in X
- It is necessary to select a value for the constant C

SVM Regression Problem

- Regression model:

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

- Loss function: $L(\varepsilon) =\ (|\varepsilon| - \delta)_+$ in contrast to $L(\varepsilon) =\ \varepsilon^2$:



- Problem statement:

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

- This problem could be solved with the change of variables and quadratic programming

SVM Regression Problem

- Change of variables:

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

- Problem statement:

$$\begin{cases} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i^+ + \xi_i^-) \to \min\limits_{w, w_0, \xi^+, \xi^-}; \\ y_i - \delta - \xi_i^- \le \langle w, x_i \rangle - w_0 \le y_i + \delta + \xi_i^+, i = 1, \dots, l; \\ \xi_i^- \ge 0, \xi_i^+ \ge 0, i = 1, \dots, l. \end{cases}$$

- This is a quadratic programming problem with linear inequality constraints. This could be solved with the dual problem (as we did in the previous section)
- Kernels are defined in a similar way to SVM classification

SVM Regression Problem

- Regression model in the dual space:

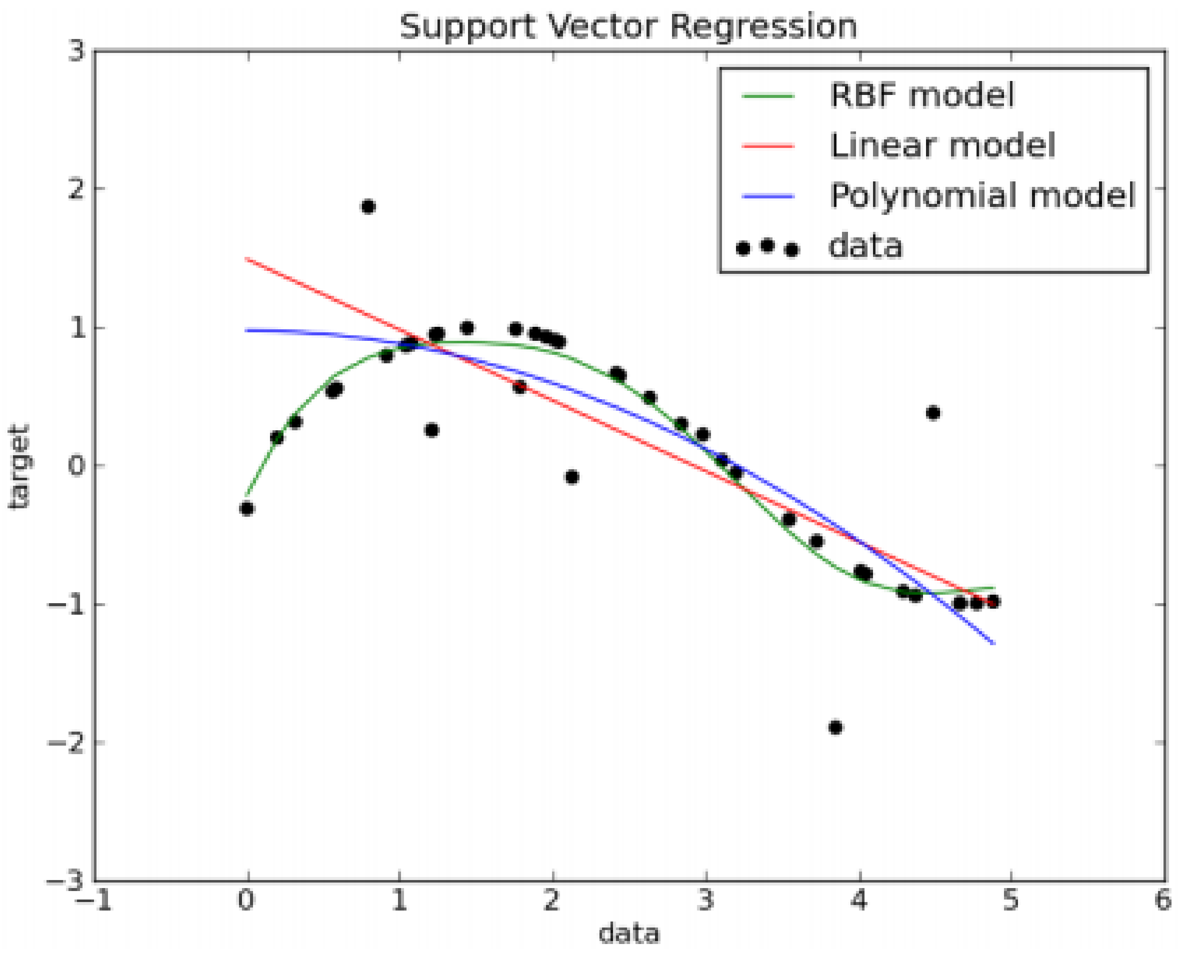
$$a(x) = \sum_{i=1}^{l}(\lambda_i^- - \lambda_i^+)K(x_i, x) - w_0;$$

where $\langle w, x_i\rangle - w_0 = \begin{cases} y_i + \varepsilon, \ if \ a(x_i) = y_i + \varepsilon, 0 < \lambda_i^+ < C, \lambda_i^- = 0, \xi_i^+ = \xi_i^- = 0; \\ y_i - \varepsilon, \ if \ a(x_i) = y_i - \varepsilon, 0 < \lambda_i^- < C, \lambda_i^+ = 0, \xi_i^+ = \xi_i^- = 0 \end{cases}$

- Parameters:
  - ε is a precision parameter
  - C is a constant (like for SVM classification)

Example from Sklearn Python:

- SVM regression with RBF kernel, linear and polynomial regression comparison:



The example is from Python sklearn
http://ogrisel.github.io/scikit-learn.org/sklearn-tutorial/auto_examples/svm/plot_svm_regression.html

SVM Regression with l1 regularization (LASSO SVM)

- Empirical risk approximation:

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

- Advantage:
  - Feature selection is done internally by choosing parameter μ: the greater the μ, the fewer the number of features

- Disadvantages:
  - LASSO starts to drop significant features when not all the insignificant features are dropped
  - There is no grouping effect: all the significant features should be selected together and have nearly equal weights w_j

SVM Regression with l1 regularization (LASSO SVM)

- Empirical risk approximation:

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

- Why l1 regularization leads to the feature selection?

Change of variables: $u_j = \frac{1}{2}(|w_j| + w_j)$, $v_j = \frac{1}{2}(|w_j| - w_j)$.

Then: $w_j = u_j - v_j \; and \; |w_j| = u_j + v_j$;

$$\begin{cases} \sum_{i=1}^{l}(1 - M_i(u - v, w_0))_+ + \mu \sum_{j=1}^{n}(u_j + v_j) \to \min_{u,v} \\ u_j \geq 0, v_j \geq 0, \qquad j = 1, \ldots, n; \end{cases}$$

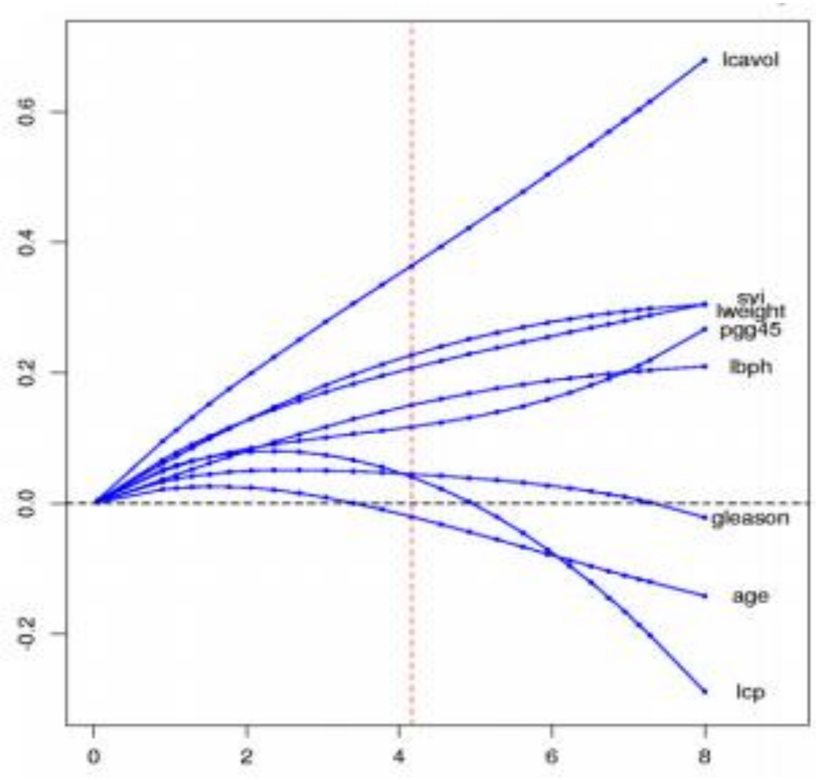
The greater the μ, the greater the number of indexes j such that $u_j = v_j = 0$,

but then w_j=0, consequently, the feature is not taken into account

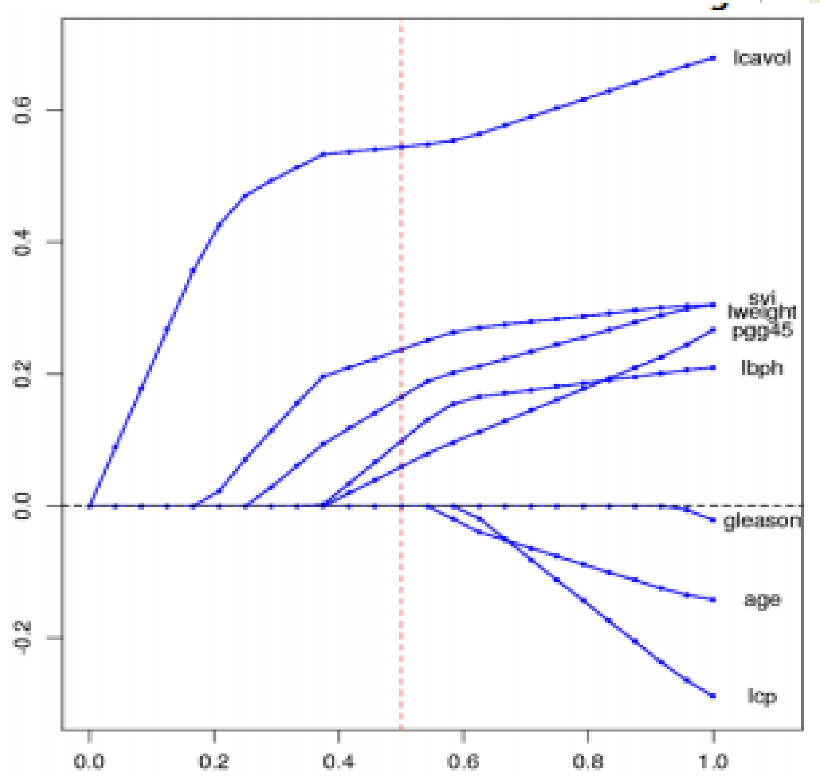
Elastic Net SVM (Doubly Regularized SVM)

- Dependence of the weights w_j from the coefficient $\frac{1}{\mu}$:

l2 regularization: $\mu \sum_j {w_j}^2$



l1 regularization: $\mu \sum_j |w_j|$



The problem is from the UCI: prostate cancer

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

l1 and l2 regularization comparison

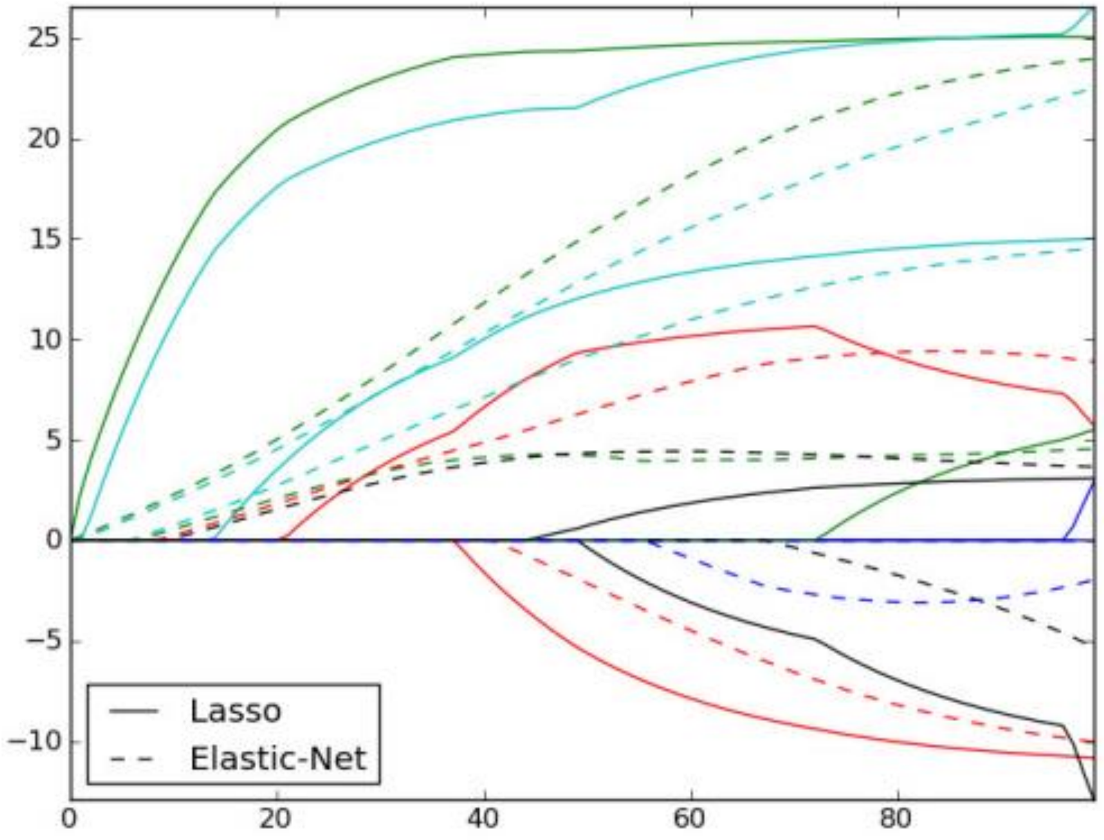
$$C\sum_{i=1}^{l}(1-M_i(w,w_0))_+ + \mu\sum_{j=1}^{n}|w_j| + \frac{1}{2}\sum_{j} {w_j}^2 \to \min_{w,w_0}$$

- Advantages:
  - Feature selection is done internally by choosing parameter μ: the greater the μ, the fewer the number of features
  - Grouping effect
- Disadvantage:
  - Noise features are also grouped together; the groups with significant features could be dropped when not all the insignificant feature groups are dropped

Li Wang, Ji Zhu, Hui Zou. The doubly regularized support vector machine // Statistica Sinica, 2006. №16, Pp. 589–615.

Elastic Net SVM

- Dependence of the weights w_j from the coefficient $\log\frac{1}{\mu}$:



The example is from Python sklearn (1.1.5)
https://scikit-learn.org/stable/modules/linear_model.html

Other regularization methods:

- Support Feature Machines (SFM)
  - 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 Feature Machines (RFM)
  - 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. 2336–233
- Bayesian Regularization
- Relevance Vector Machines (RVM)

Summary:

- SVM is the best linear classification method
- SVM has beautiful generalization both for non-linear classification and linear and non-linear regression
- Threshold loss function approximation makes the gap greater and, consequently, makes the overall performance higher
- Regularization eliminates multicollinearity and overfitting
- Regularization is equivalent to the definition of the a-priory distribution in the coefficient space
- l1 and the other non-standard regularizations makes internal feature selection without the explicit search of subsets