Exercise in RPZ
Support Vector Machines

Vojtěch Franc, Martin Švec

November 13, 2009

1 Introduction

The aim of this exercise is to implement learning algorithm called Support Vector Machines (SVM). SVM is algorithm that realizes the principle of structural risk minimization. The problem of structural risk minimization is converted to maximization distance between separating hyperplane and the closest examples from the training set - this distance is called margin. Finally, the problem of margin maximization is converted to minimizing quadratic criterion. If we express this criterion in such a way that the criterion contains points from the training set only inside scalar product, we can use kernel functions to generalize separating hyperplane onto nonlinear classifier (nonlinear separating hyperplane).

2 Support Vector Machines

Linear SVM search for hyperplane $\langle w, x\rangle + b = 0$, which maximizes distance from the points in linear separable training set $\{(x_1, y_1), (x_2, y_2), ..., (x_l, y_l)\}$, where $x_i \in \mathbb{R}^n$ is training sample and $y_i \in \{+1, -1\}$ his class identifier (label). The problem of finding hyperplane that maximizes the distance from classes is equivalent to solving optimization task

$$(w, b) = \operatorname*{argmin}_{w,b} \frac{1}{2}||w||^2 \tag{1}$$

under conditions

$$\begin{array}{rcll} \langle w, x_i\rangle + b & \geq & +1\,, & y_i = +1\,, \\ \langle w, x_i\rangle + b & \leq & -1\,, & y_i = -1\,. \end{array} \tag{2}$$

If we solve task (1), we get vector w and threshold b that determine separating hyperplane. However, it is advantageous to revert the equation onto so-called dual task instead of solving it directly. The aim of dual task is to find numbers $alpha_i$, $i = 1, ..., l$, which are solution of the task

$$\vec{\alpha} = \underset{\vec{\alpha}}{\operatorname{argmax}} \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \;, \tag{3}$$

under condition

$$\begin{aligned} \alpha_i &\geq 0 \,, \quad i = 1, 2, ..., l \,, \\ \sum_{i=1}^{l} \alpha_i y_i &= 0 \,. \end{aligned}$$

The numbers α_i determine separating hyperplane, i.e. its normal vector w and threshold b, that can be determined in a following way

$$\begin{aligned} w &= \sum_{i=1}^{l} \alpha_i y_i x_i \;, \\ b &= \frac{1}{|I|} \sum_{i \in I} (y_i - \sum_{j=1}^{l} \alpha_j y_j \langle x_j, x_i \rangle) \;, \end{aligned}$$

where the set I contains indices of points for which $0 < \alpha < C$. The advantage of dual task is that instead of optimizing vector w, we can just optimize real numbers α_i under substantial easier constraints than it was in primary task. The other advantage is the fact that points x_i occur in dual task in the form of scalar products. This enables us to use kernel functions for seeking nonlinear classifier instead of separating hyperplane. The whole modification consist in replacing all scalar products $\langle x_i, x_j \rangle$ by appropriate kernel function $k(x_i, x_j)$. Classification of point x is done according to a sign of function

$$f(x) = \sum_{i=1}^{l} \alpha_i y_i k(x_i, x_j) + b \;,$$

which is nothing else than decision about on which side of separating hyperplane $\langle w, \phi(x) \rangle + b = 0$ in a linearized feature space the point x lies. It means that we get *nonlinear SVM* using substituting scalar products by kernel functions $k(x_i, x_j)$.

For the primary task (1) we supposed separability of the training set, it means that there can be hyperplane dividing training set with zero error. In other words, all points from the first class $\{x_i \colon y_i = +1\}$ have positive distance from separating hyperplane and points from the second class $\{x_i \colon y_i = -1\}$ have negative distance form the hyperplane. This requirement is expressed by system of inequalities (2) of the primary task. For nonseparable training set, the hyperplane that would divide the data with zero error does not exist and therefore the system of inequalities (2) does not have solution. Solution for this case are SVM for nonseparable data. It consists in making constraints less tight by adding new relaxation variables ξ_i. Primary task has than the form

$$(w, b, \xi_i) = \operatorname*{argmin}_{w,b,\xi_i} \frac{1}{2}||w||^2 + C \sum_{i=1}^{l} \xi_i \,, \tag{4}$$

under constraints

$$\begin{array}{rcll} \langle w, x_i \rangle + b & \geq & +1 - \xi_i \,, & y_i = +1 \,, \\ \langle w, x_i \rangle + b & \leq & -1 + \xi_i \,, & y_i = -1 \,, \end{array} \tag{5}$$

We see in criterion (4) that there is new term $C \sum_{i=1}^{l} \xi_i$, which express the demand for having as small number of points that violate the condition (5) as it is possible. Constant C (chosen before optimization) determines whether we prefer minimization of the first term (margin maximization) or minimization of the second term (number of points violating system of constraints). Let us note that for separable case we can set $C = \infty$, since the hyperplane for zero violating points can be found. For this case, we can also use dual task. It has the same for as dual task for separable case, having one more constraint $0 \leq \alpha_i \leq C$. Then, the dual task for SVM for nonseparable case is

$$\vec{\alpha} = \operatorname*{argmax}_{\vec{\alpha}} \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \,, \tag{6}$$

under condition

$$\begin{array}{rcl} C \geq \alpha_i & \geq & 0 \,, \quad i = 1, 2, ..., l \,, \\ \sum_{i=1}^{l} \alpha_i y_i & = & 0 \,. \end{array}$$

To solve dual task, use the function `gsmo` from the STPR toolbox. This function requires formulating the quadratic criterion using matrices. Dual task (6) can be stated using matrices as

$$\vec{\alpha} = \operatorname*{argmax}_{\vec{\alpha}} \vec{\alpha}^T \cdot \vec{1} - \frac{1}{2} \vec{\alpha}^T \cdot H \cdot \vec{\alpha} \,,$$

under conditions

$$\begin{array}{rcl} \vec{\alpha}^T \cdot \vec{y} & = & \vec{0} \,, \\ \vec{0} \leq \vec{\alpha} & \leq & \vec{C} \,, \end{array}$$

where

$\vec{\alpha}$	is vector $[l \times 1]$ of seeked numbers,
H	is matrix $[l \times l]$ and its elements are $H(i,j) = y_i y_j k(x_i, x_j)$,
$\vec{C}$	is vector $[l \times 1]$ containing C,
$\vec{y}$	is vector $[l \times 1]$ containing class identifiers y_i,
$\vec{1}$	is vector $[l \times 1]$ containing ones,
$\vec{0}$	is vector $[l \times 1]$ containing zeros.