Datamining 2 - Jennifer Probst - HW 6

Ex 1 a) for this variant of SVM (transductive SVM) we formulate the Lagrangean:

$$L((w, b, \xi, \xi_-, \xi_+), \alpha, \alpha^*, \beta, \beta^*) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i + C_-^*\sum_{j:y_j^*=-1}\xi_j^* + C_+^*\sum_{j:y_j^*=1}\xi_j^*$$
$$- \sum_{i=1}^{n}\alpha_i\left(y_i(\langle\vec{w},\vec{x_i}\rangle + b) - 1 + \xi_i\right) - \sum_{i=1}^{n}\beta_i\xi_i$$
$$- \sum_{j:y_j^*=-1}\alpha_j^*\left(y_j^*(\langle\vec{w},\vec{x_j^*}\rangle + b) - 1 + \xi_j^*\right) - \sum_{j:y_j^*=-1}\beta_j^*\xi_j^*$$
$$- \sum_{j:y_j^*=1}\alpha_j^*\left(y_j^*(\langle\vec{w},\vec{x_j^*}\rangle + b) - 1 + \xi_j^*\right) - \sum_{j:y_j^*=1}\beta_j^*\xi_j^*$$

we now maximizing the Lagrangian with respect to w, b, ξ, ξ_-, ξ_+ taking the first derivatives and setting them to zero.

$$\frac{\partial L}{\partial b} \overset{!}{=} 0 = -\sum_{i=1}^{n}\alpha_i y_i - \sum_{j:y_j^*=-1}\alpha_j^* y_j^* - \sum_{j:y_j^*=1}\alpha_j^* y_j^* \quad (*)$$
$$= \sum_{i=1}^{n}\alpha_i y_i + \sum_{i=n+1}^{k}\alpha_i y_i$$

$$\boxed{0 = \sum_{i=1}^{n+k}\alpha_i y_i}$$

(*) by setting the indices j of the k test points to indices n+1 to n+k we can derive a more compact form where the α_i with $i \in (n+1, n+k)$ are weights belonging to y_i with $i \in (n+1, n+k)$ from the original test set.

$$\frac{\partial L}{\partial \vec{w}} \overset{!}{=} 0 = \vec{w} - \sum_{i=1}^{n}\alpha_i y_i \vec{x_i} - \sum_{j:y_j^*=-1}\alpha_j^* y_j^* \vec{x_j^*} - \sum_{j:y_j^*=1}\alpha_j^* y_j^* \vec{x_j^*}$$

(*) simplified with same argument as above

$$\vec{w} = \sum_{i=1}^{n}\alpha_i y_i \vec{x_i} + \sum_{i=n+1}^{k}\alpha_i y_i \vec{x_i}$$

$$\boxed{\vec{w} = \sum_{i=1}^{n+k}\alpha_i y_i \vec{x_i}}$$

$$\frac{\partial L}{\partial \xi_i} \overset{!}{=} 0 = C - \alpha_i - \beta_i \Rightarrow \boxed{\beta_i = C - \alpha_i}$$

with the constraint $\beta_i \geq 0$
$\rightarrow C - \alpha_i \geq 0 \rightarrow \alpha_i \leq C$
$\Rightarrow 0 \leq \alpha_i \leq C \quad \forall i \in \{1, \dots, n\}$

$$\frac{\partial L}{\partial \xi_{-j}} \overset{!}{=} 0 = C_-^* - \alpha_j^* - \beta_j^* \Rightarrow \boxed{\beta_j^* = C_-^* - \alpha_j^*}$$

with $\beta_j^* \geq 0 \rightarrow C_-^* - \alpha_j^* \geq 0 \rightarrow \alpha_j^* \leq C_-^*$
(*) $\Rightarrow 0 \leq \alpha_i \leq C_-^* \quad \forall i \in \{n+1, \dots, n+k\}$ with $y_i = -1$

$$\frac{\partial L}{\partial \xi_{+j}} \overset{!}{=} 0 = C_+^* - \alpha_j^* - \beta_j^* \Rightarrow \boxed{\beta_j^* = C_+^* - \alpha_j^*}$$

with $\beta_j^* \geq 0 \rightarrow C_+^* - \alpha_j^* \geq 0 \rightarrow \alpha_j^* \leq C_+^*$

(*) $\Rightarrow 0 \leq \alpha_i \leq C_+^* \quad \forall i \in \{n+1, \dots, n+k\}$ for $y_i = 1$

substituting these findings in the lagrangian we can see that it simplifies a lot:

already using the indexing (*))

$$\max_{\alpha} L(\alpha) = +C\sum_{i=1}^{n+k}\xi_i - \sum_{i=1}^{n+k}\beta_i\xi_i - \sum_{i=1}^{n+k}\alpha_i\left(y_i(\langle\vec{w},\vec{x}_i\rangle+b)-1+\xi_i\right) + \frac{1}{2}\|w\|^2$$

$$\left(\beta_i = C-\alpha_i,\ \sum_{i=1}^{n+k}\alpha_i y_i = 0\right) = +C\sum_{i=1}^{n+k}\xi_i - \sum_{i=1}^{n+k}(C-\alpha_i)\xi_i - \sum_{i=1}^{n+k}\alpha_i y_i(\langle\vec{w},\vec{x}_i\rangle+b) + \sum_{i=1}^{n+k}\alpha_i - \sum_{i=1}^{n+k}\alpha_i\xi_i + \frac{1}{2}\|w\|^2$$

$$= +C\sum_{i=1}^{n+k}\xi_i - C\sum_{i=1}^{n+k}\xi_i + \sum_{i=1}^{n+k}\alpha_i\xi_i - b\sum_{i=1}^{n+k}\alpha_i y_i - \sum_{i=1}^{n+k}\alpha_i\xi_i + \sum_{i=1}^{n+k}\alpha_i - \sum_{i=1}^{n+k}\alpha_i y_i(\langle\vec{w},\vec{x}_i\rangle+b) + \frac{1}{2}\|w\|^2$$

$$\left(\text{using } w=\sum_{i=1}^{n+k}\alpha_i y_i\vec{x}_i\right) = \sum_{i=1}^{n+k}\alpha_i + \frac{1}{2}\left\langle\sum_{i=1}^{n+k}\alpha_i y_i\vec{x}_i, \sum_{j=1}^{n+k}\alpha_j y_j\vec{x}_j\right\rangle - \left\langle\sum_{i=1}^{n+k}\alpha_i y_i x_i, w\right\rangle$$

$$= \sum_{i=1}^{n+k}\alpha_i - \frac{1}{2}\sum_{i=1}^{n+k}\sum_{j=1}^{n+k}\alpha_i\alpha_j y_i y_j\langle\vec{x}_i,\vec{x}_j\rangle$$

→ The dual representation is:

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{n+k}\alpha_i - \frac{1}{2}\sum_{i=1}^{n+k}\sum_{j=1}^{n+k}\alpha_i\alpha_j y_i y_j\langle\vec{x}_i,\vec{x}_j\rangle$$

s.t. $\sum_{i=1}^{n+k}\alpha_i y_i = 0$ with

$0 \le \alpha_i \le C \quad \forall i \in \{1,\dots,n\}$

$0 \le \alpha_i \le C_-^* \quad \forall i \in \{n+1,\dots,n+k\}$ and $y_i = -1$

$0 \le \alpha_i \le C_+^* \quad \forall i \in \{n+1,\dots,n+k\}$ and $y_i = 1$

□

2c) ∘ A Hilbert space H is: ∘ a vector space over real numbers $\mathbb{R}$ which has a norm $\|\cdot\|_H$ and which is complete in that norm.
∘ In addition the norm comes from an inner product $\langle\cdot,\cdot\rangle_H$ (i.e. $\|f\|_H = \sqrt{\langle f,f\rangle_H}$)

- A kernel is a function $K: X\times X \to \mathbb{R}$ if there exists a Hilbert space H over $\mathbb{R}$ and the map $\Phi: X \to H$ such that $\forall x,y \in X: K(x,y) = \langle\Phi(x),\Phi(y)\rangle_H$
- The evaluation functional at $x \in X$ in H is defined as the map $L_x: H \to \mathbb{R}$ such that for all functions $f \in H$ it holds: $L_x(f) = f(x)$

→ we then call H a reproducing kernel Hilbert space if $\forall x \in X$, L_x is bounded (∃ M constant and finite) such that $\|L_x(f)\| = |f(x)| \le M\|f\|_H$

→ This reproducing kernel Hilbert space is associated with a kernel where evaluation at any x from every function in the space is reproduced by taking an inner product with a function determined by the kernel:

↳ According to the Riesz representation theorem: $\forall x \in X$ ∃ some function $K_x \in H$ such that $f(x) = L_x(f) = \langle f, K_x\rangle_H \quad \forall f \in H$.

But as K_x is also a function on H, for any $y \in X$: $K_x(y) = L_y(K_x) = \langle K_x, K_y\rangle_H$

→ if we now define $K: X\times X \to \mathbb{R}$: $K(x,y) = \langle K_x, K_y\rangle_H$

with $\Phi(x) = K_x$ → the kernel reproduces the inner product $K(x,y) = \langle\Phi(x),\Phi(y)\rangle_H$

2d) I would choose them by using cross validation for C_1 and C_2. C_1 would probably be larger than C_2, as the label for test data set is not known for certain → small C_2. For the training data I would try a range of values for C_1 in the cross validation (only use part of ds to train & predict) to get a trade off btw misclassification & margin size.