CSE 417T: Introduction to Machine Learning

Lecture 22: The Kernel Trick

Henry Chai
11/15/18
Linearly Inseparable Data

- What can we do if the data is not linearly separable?
  - Accept some non-zero in-sample error
  - How much in-sample error should we tolerate?
- Apply a non-linear transformation that shifts the data into a space where it is linearly separable
  - How can we pick a non-linear transformation?
minimize \( \frac{1}{2} \mathbf{w}^T \mathbf{w} \)

subject to \( y_i (\mathbf{w}^T \mathbf{x}_i + w_0) \geq 1 \quad \forall (\mathbf{x}_i, y_i) \in \mathcal{D} \)

• When \( \mathcal{D} \) is not linearly separable, there are no feasible solutions to this optimization problem
Soft-Margin SVMs

minimize \( \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^{n} \xi_i \)

subject to \( y_i (\mathbf{w}^T \mathbf{x}_i + w_0) \geq 1 - \xi_i \) \( \forall (\mathbf{x}_i, y_i) \in \mathcal{D} \)

\( \xi_i \geq 0 \) \( \forall i \in \{1, \ldots, n\} \)

- \( \xi_i \) is the “soft” error on the \( i^{th} \) training
  - If \( \xi_i > 1 \), then \( y_i (\mathbf{w}^T \mathbf{x}_i + w_0) < 0 \) \( \Rightarrow (\mathbf{x}_i, y_i) \) is incorrectly classified
  - If \( 0 < \xi_i < 1 \), then \( y_i (\mathbf{w}^T \mathbf{x}_i + w_0) > 0 \) \( \Rightarrow (\mathbf{x}_i, y_i) \) is correctly classified but inside the margin

- \( \sum_{i=1}^{n} \xi_i \) is the “soft” in-sample error
Primal-Dual Optimization

**Primal**

\[
\text{minimize} \quad \frac{1}{2} \mathbf{w}^T \mathbf{w} \\
\text{subject to} \quad y_i (\mathbf{w}^T \mathbf{x}_i^T + w_0) \geq 1 \quad \forall (\mathbf{x}_i, y_i) \in \mathcal{D}
\]

**Dual**

\[
\text{minimize} \quad \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^{n} \alpha_i \\
\text{subject to} \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \\
\alpha_i \geq 0 \quad \forall \ i \in \{1, \ldots, n\}
\]
Primal-Dual Optimization

- **Primal**
  - Directly returns the hyperplane, \([w_0^*, \bar{w}^*]\)
  - Support vectors are all \((\bar{x}_s, y_s) \in \mathcal{D}\) s.t. \(y_s \left(\bar{w}^*^T \bar{x}_s + w_0^*\right) = 1\)

- **Dual**
  - Returns the vector, \(\bar{\alpha}^*\)
    \[
    \bar{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i \bar{x}_i
    \]
    \[
    w_0^* = y_s - \bar{w}^*^T \bar{x}_s \quad \text{where} \quad \alpha_i^* > 0
    \]
  - Support vectors are all \((\bar{x}_s, y_s) \in \mathcal{D}\) s.t. \(\alpha_i > 0\)
Primal-Dual Optimization

• Primal
  \[ g(\bar{x}) = \text{sign}(\bar{w}^T \bar{x} + w_0^*) \]

• Dual
  \[ g(\bar{x}) = \text{sign}(\bar{w}^T \bar{x} + w_0^*) \]
  \[ = \text{sign} \left( \sum_{\ell : u_i > 0} \alpha_i^* y_i \bar{x}_i^\tau \cdot \bar{x} + w_0^* \right) \]
Primal-Dual Soft-Margin SVMs

Primal

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^{n} \xi_i \\
\text{subject to} & \quad y_i (\mathbf{w}^T \mathbf{x}_i + w_0) \geq 1 - \xi_i \quad \forall (\mathbf{x}_i, y_i) \in \mathcal{D} \\
& \quad \xi_i \geq 0 \quad \forall i \in \{1, \ldots, n\}
\end{align*}
\]

Dual

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^{n} \alpha_i \\
\text{subject to} & \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \\
& \quad 0 \leq \alpha_i \leq C \quad \forall i \in \{1, \ldots, n\}
\end{align*}
\]
The Lagrangian for this soft-margin SVM now has even more new variables: specifically we have a new variable for each constraint on the $\xi_i$’s. We’re going to call these new variables $\beta_i$
When we look at minimizing this Lagrangian, the only new thing we have to consider is the gradient w.r.t. the new variables $\xi_i$. 

\[
\begin{align*}
\text{minimize} & \quad L(\hat{a}, \hat{b}, \hat{w}, w_0, \xi) \\
L(\hat{a}, \hat{b}, \hat{w}, w_0, \xi) &= \frac{1}{2} \hat{w}^T \hat{w} + C \sum_{i=1}^{n} \xi_i \\
&\quad + \sum_{i=1}^{n} \alpha_i (1 - \xi_i - y_i (\hat{w}^T \tilde{x}_i + w_0)) - \sum_{i=1}^{n} \beta_i \xi_i \\
\frac{\partial L(\hat{a}, \hat{b}, \hat{w}, w_0, \xi)}{\partial \hat{w}} &= \hat{w} - \sum_{i=1}^{n} \alpha_i y_i \tilde{x}_i \rightarrow \tilde{w}^* = \sum_{i=1}^{n} \alpha_i y_i \tilde{x}_i \\
\frac{\partial L(\hat{a}, \hat{b}, \hat{w}, w_0, \xi)}{\partial w_0} &= - \sum_{i=1}^{n} \alpha_i y_i \rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0 \\
\frac{\partial L(\hat{a}, \hat{b}, \hat{w}, w_0, \xi)}{\partial \xi_i} &= C - \alpha_i - \beta_i \rightarrow \beta_i = C - \alpha_i \forall \ i
\end{align*}
\]
This last form for the minimum of the Lagrangian is the same as the minimum of the Lagrangian for the hard-margin SVM as all of the new variables that we introduced in the soft-margin \((C, \beta_i, \xi^*_i)\) have just disappeared from the final form! Note that we have a new constraint though, the \(\beta_i = C - \alpha_i \ \forall \ i\).
We can remove the constraints that are affiliated with $\beta_i$ (as $\beta_i$ does not appear in the objective) by incorporating them into the constraint on $\alpha_i$. 

$$\text{maximize} \quad -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \overline{x_i}^T \overline{x_j} + \sum_{i=1}^{n} \alpha_i$$

subject to $\sum_{i=1}^{n} \alpha_i y_i = 0$

$\alpha_i \geq 0 \quad \forall \ i \in \{1, ..., n\}$

$\beta_i \geq 0 \quad \forall \ i \in \{1, ..., n\}$

$\beta_i = c - \alpha_i \quad \forall \ i \in \{1, ..., n\}$

We can remove the constraints that are affiliated with $\beta_i$ (as $\beta_i$ does not appear in the objective) by incorporating them into the constraint on $\alpha_i$. 

$$\text{maximize} \quad -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \overline{x_i}^T \overline{x_j} + \sum_{i=1}^{n} \alpha_i$$

subject to $\sum_{i=1}^{n} \alpha_i y_i = 0$

$0 \leq \alpha_i \leq c \quad \forall \ i \in \{1, ..., n\}$
Primal-Dual Soft-Margin SVMs

- Primal
  - Directly returns the hyperplane, \([w_0^*, \overline{w}^*]\)
  - Support vectors are all \((x_s^i, y_s^i) \in D\) s.t. \(y_s^i (\overline{w}^* \overline{x}_s^i + w_0^i) = 1\)

- Dual
  - Returns the vector, \(\overline{a}^*\)

\[
\overline{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i \overline{x}_i^i
\]

\(w_0^* = ???\)
So in order for these two optimization problems to be equivalent, all of the excess stuff in the objective function must go to zero at optimality. Note that each term in both of the two sums in the second line will be non-positive i.e. each term will always be \( \leq 0 \) for any feasible setting of \( \vec{w}, w_0, \vec{\xi} \) so none of the terms in the sum can like cancel out other terms (i.e. there are some positive and some negative terms). Therefore, we can conclude that each individual term inside both of the sums must be zero.
minimize \( \frac{1}{2} \overline{w}^T \overline{w} + C \sum_{i=1}^{n} \xi_i \)
subject to \( 1 - \xi_i - y_i (\overline{w}^T \overline{x}_i + w_0) \leq 0 \ \forall (\overline{x}_i, y_i) \in D \)
\( -\xi_i \leq 0 \quad \forall i \in \{1, \ldots, n\} \)

\[
\uparrow
\]

maximize \( \min \frac{\alpha}{\overline{w}, w_0, \xi} \quad L(\alpha, \overline{w}, \overline{w}, w_0, \xi) \)

\( \alpha, \overline{w}, w_0, \xi \geq 0 \)

- Theorem: \( \alpha_i^* (1 - \xi_i^* - y_i (\overline{w}_i^T \overline{x}_i + w_0^*)) = 0 \ \forall (\overline{x}_i, y_i) \in D \)

and \( \beta_i^* \xi_i^* = (C - \alpha_i^*) \xi_i^* = 0 \ \forall (\overline{x}_i, y_i) \in D \)

- If \( 0 < \alpha_s^* \), then \( 1 - \xi_s^* - y_s (\overline{w}_s^T \overline{x}_s + w_0^*) = 0 \)
- If \( \alpha_s^* < C \), then \( \xi_s^* = 0 \)
- If \( 0 < \alpha_s^* < C \), then \( 1 - y_s (\overline{w}_s^T \overline{x}_s + w_0^*) = 0 \)
Complementary Slackness

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^{n} \xi_i \\
\text{subject to} & \quad 1 - \xi_i - y_i (\mathbf{w}^T \mathbf{x}_i + w_0) \leq 0 \quad \forall (\mathbf{x}_i, y_i) \in \mathcal{D} \\
& \quad -\xi_i \leq 0 \quad \forall i \in \{1, \ldots, n\}
\end{align*}
\]

\[\begin{align*}
\uparrow
\end{align*}\]

\[
\begin{align*}
\text{maximize} \quad & \minimize \quad L(\bar{a}, \bar{b}, \bar{w}, w_0, \bar{\xi}) \\
\text{subject to} \quad & \bar{a}, \bar{b} \geq 0 \quad \minimize \quad L(\bar{a}, \bar{b}, \bar{w}, w_0, \bar{\xi}) \\
\end{align*}
\]

\[
\begin{align*}
& \text{Theorem: } \alpha_i^* \left(1 - \xi_i^* - y_i \left(\mathbf{w}_i^* \mathbf{x}_i + w_0^*\right)\right) = 0 \quad \forall (\mathbf{x}_i, y_i) \in \mathcal{D} \\
& \quad \text{and } \beta_i^* \xi_i^* = (C - \alpha_i^*) \xi_i^* = 0 \quad \forall (\mathbf{x}_i, y_i) \in \mathcal{D} \\
& \quad \text{If } 0 < \alpha_s^*, \text{ then } 1 - \xi_s^* - y_s \left(\mathbf{w}_s^* \mathbf{x}_s + w_0^*\right) = 0 \\
& \quad \text{If } \alpha_s^* < C, \text{ then } \xi_s^* = 0 \\
& \quad \text{If } 0 < \alpha_s^* < C, \text{ then } w_0^* = y_s - \mathbf{w}_s^* \mathbf{x}_s
\end{align*}
\]
Primal-Dual Soft-Margin SVMs

- Primal
  - Directly returns the hyperplane, \([w_0^*, \bar{w}^*]\)
  - Support vectors are all \((x_s^i, y_s^i) \in D\) s.t. \(y_s^i (\bar{w}^* x_s^i + w_0^*) = 1\)

- Dual
  - Returns the vector, \(\alpha^*\)
    \[
    \bar{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i
    \]
    \[
    w_0^* = y_s^i - \bar{w}^* x_s^i \text{ where } 0 < \alpha_s^* < C
    \]
  - Support vectors are all \((x_s^i, y_s^i) \in D\) s.t. \(0 < \alpha_s^* < C\)
  - If \(\alpha_s^* = C\), then \(\xi_s^i > 0 \Rightarrow (x_s^i, y_s^i)\) is inside the margin or misclassified
Assume for now that you’ve lucked into a transformation that makes the data linearly separable so we can use a hard-margin SVM.
As we’ve now seen, instead of solving the primal we can equivalently solve the dual problem.

Nonlinear Dual SVMs

- Decide on a transformation $\Phi: \mathcal{X} \rightarrow \mathcal{Z}$

- Find a maximal-margin separating hyperplane in the transformed space, $[\overline{w}^*, \overline{w}_0^*]$, by solving the QP:

$$\min \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \overline{a}_i \overline{a}_j y_i y_j \Phi(\overline{x}_i^*)^T \Phi(\overline{x}_j^*) - \sum_{i=1}^{n} \overline{a}_i$$

subject to $\sum_{i=1}^{n} \overline{a}_i y_i = 0$

$\overline{a}_i \geq 0 \ \forall \ i \in \{1, ..., n\}$

- Return the corresponding predictor in the original space:

$$g(\tilde{x}) = \text{sign}\left(\sum_{i: \overline{a}_i > 0} \overline{a}_i^* y_i \Phi(\overline{x}_i^*)^T \Phi(\tilde{x}) + \overline{w}_0^*\right)$$
First the good news: when we originally introduced non-linear transformations, we had this problem of overfitting where we would be able to drive our in-sample error to zero but our out-of-sample error would explode. This is no longer the case with SVMs.
Efficiency

• Depending on the transformation $\Phi$ and the dimensionality of the original input space $X$, computing $\Phi(\tilde{x})$ can be computationally expensive
  • Computing $\Phi_2(\tilde{x}) = [x_1, x_2, \ldots, x_D, x_1^2, x_1x_2, \ldots, x_D^2]$ requires $\bar{D} = D + \binom{D}{2} + D = \frac{D^2 + 3D}{2} = O(D^2)$ time
  • Computing $\Phi_{10}(\tilde{x})$ requires $O(D^{10})$ time

• Tradeoff:
  • High-dimensional transformations can result in good hypotheses (as long as they don’t overfit)
  • High-dimensional transformations are expensive
The Dual SVM method is known as an inner product algorithm because the input vectors $\mathbf{x}_i$ only appear as inner products with other input vectors (including in $\omega_0^*$).

Nonlinear Dual SVMs

- Insight: the transformation $\Phi$ only appears as the inner product between two transformed points
- Find a maximal-margin separating hyperplane in the transformed space, $[\tilde{w}^*, \tilde{\omega}_0^*]$, by solving the QP:

$$
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \tilde{a}_i \tilde{a}_j y_i y_j \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j) - \sum_{i=1}^{n} \tilde{a}_i \\
\text{subject to} & \quad \sum_{i=1}^{n} \tilde{a}_i y_i = 0 \\
& \quad \tilde{a}_i \geq 0 \quad \forall \ i \in \{1, ..., n\}
\end{align*}
$$

- Return the corresponding predictor in the original space:

$$
g(\mathbf{x}) = \text{sign} \left( \sum_{i: \tilde{a}_i^* > 0} \tilde{a}_i^* y_i \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}) + \tilde{\omega}_0^* \right)
$$
The Kernel Trick

- Approach: instead of computing $\Phi(\tilde{x})$, find a function $K_\Phi$ s.t. $K_\Phi(\tilde{x}, \tilde{x}') = \Phi(\tilde{x})^T \Phi(\tilde{x}') \forall \tilde{x}, \tilde{x}' \in \mathcal{X}$.
  - $K_\Phi(\tilde{x}, \tilde{x}')$ should be cheaper to compute than $\Phi(\tilde{x})$.

- Example:
  
  $\Phi_2'(\tilde{x}) = [x_1, x_2, ..., x_D, x_1^2, \sqrt{2}x_1x_2, ..., \sqrt{2}x_{D-1}x_D, x_D^2]$

  
  $\Phi_2'(\tilde{x})^T \Phi_2'(\tilde{x}') = \sum_{i=1}^D x_i x_i' + \sum_{i=1}^D x_i^2 x_i'^2 + \sum_{i=1}^D \sum_{j \neq i}^D 2x_i x_i' x_j x_j'$

  
  
  $= \sum_{i=1}^D x_i x_i' + \left( \sum_{i=1}^D x_i x_i' \right)^2$

  
  $K_{\Phi_2'}(\tilde{x}, \tilde{x}') = \tilde{x}^T \tilde{x}' + (\tilde{x}^T \tilde{x}')^2$

Note that $\Phi_2'$ is different from $\Phi_2$ but the two transformations are equivalently expressive.
Computing the kernel is an order of magnitude faster than directly computing the transformation.

- **Approach:** instead of computing $\Phi(\tilde{x})$, find a function $K_\Phi$ s.t. $K_\Phi(\tilde{x}, \tilde{x}') = \Phi(\tilde{x})^T \Phi(\tilde{x}') \forall \tilde{x}, \tilde{x}' \in X$
  - $K_\Phi(\tilde{x}, \tilde{x}')$ should be cheaper to compute than $\Phi(\tilde{x})$

- **Example:**
  
  $\Phi'_2(\tilde{x}) = [x_1, x_2, ..., x_D, x_1^2, \sqrt{2}x_1x_2, ..., \sqrt{2}x_{D-1}x_D, x_D^2]$

  
  $\Phi'_2(\tilde{x})^T \Phi'_2(\tilde{x}') = \tilde{x}^T \tilde{x}' + (\tilde{x}^T \tilde{x}')^2 = K_{\Phi'_2}(\tilde{x}, \tilde{x}')$

  - Computing $\Phi'_2(\tilde{x})^T \Phi'_2(\tilde{x}')$ requires $O(D^2)$ time whereas computing $K_{\Phi'_2}(\tilde{x}, \tilde{x}')$ only takes $O(D)$
\[ K_{\Phi_2}'(\tilde{x}, \tilde{x}') = \tilde{x}^T \tilde{x}' + (\tilde{x}^T \tilde{x}')^2 \]

- Implied feature transformation: \( \Phi_2' (\tilde{x}) = [x_1, x_2, ..., x_D, x_1^2, \sqrt{2}x_1x_2, ..., \sqrt{2}x_{D-1}x_D, x_D^2] \)
- Implied dimensionality: \( \tilde{D} = \frac{D^2 + 3D}{2} \)

\[ K_{\Phi_2(\zeta, \gamma)} (\tilde{x}, \tilde{x}') = (\zeta + \gamma \tilde{x}^T \tilde{x}')^2 - \zeta^2 \]

- Implied feature transformation: \( \Phi_2(\zeta, \gamma) (\tilde{x}) = [\sqrt{2}\zeta x_1, ..., \sqrt{2}\zeta x_D, \gamma x_1^2, \gamma x_1x_2, ..., \gamma x_D^2] \)
- Implied dimensionality: \( \tilde{D} = \frac{D^2 + 3D}{2} \)

- \( \gamma \) and \( \zeta \) affect the geometry of the transform but not the expressiveness

\[ \gamma \text{ and } \zeta \text{ affect the geometry of the transform but not the expressiveness} \]
The implied features of the RBF Kernel are kind of weird looking but if you squint kind of hard, you can see that they all powers of each dimension of the original input are in there, just weighted in a decaying fashion. It comes from the Taylor series for the exponential function
• Polynomial Kernel: \( K_{\phi(\zeta, \gamma)}(\vec{x}, \vec{x}') = (\zeta + \gamma \vec{x}^T \vec{x}')^Q - \zeta^Q \)
  • Implied dimensionality: \( \tilde{D} = O(D^Q) \)
  • \( \gamma \) and \( \zeta \) affect the geometry of the transform: changing them changes the support vectors / decision boundary
    • Set using validation

• Gaussian-RBF Kernel: \( K_{\phi_r}(\vec{x}, \vec{x}') = e^{-\frac{\|\vec{x} - \vec{x}'\|^2}{2r}} \)
  • Implied feature transformation: \( \Phi_r(\vec{x}) = \left[ e^{-\frac{x_1^2}{2r}}, \sqrt{\frac{\pi^d}{d!}} \sum_{i=1}^{d} \sqrt{\frac{x_i^2}{2r}}, \ldots, e^{-\frac{x_d^2}{2r}}, \sqrt{\frac{\pi^d}{d!}} \sum_{i=1}^{d} \sqrt{\frac{x_i^2}{2r}} \right] : d \in \mathbb{N} \)
  • Implied dimensionality: \( \tilde{D} = \infty \)
  • Set \( r \) using validation
• Any function $K$ is a valid kernel if and only if:
  • $\exists$ a transformation $\Phi$ s.t. $K(x, x') = \Phi(x)^T \Phi(x')$ $\forall$ $x, x'$

$\Phi$

• the Gram matrix

$$
K = \begin{bmatrix}
K(x_1, x_1) & K(x_1, x_2) & \cdots & K(x_1, x_n) \\
K(x_2, x_1) & K(x_2, x_2) & \cdots & K(x_2, x_n) \\
\vdots & \vdots & \ddots & \vdots \\
K(x_n, x_1) & K(x_n, x_2) & \cdots & K(x_n, x_n)
\end{bmatrix}
$$

is positive semi-definite $\forall$ sets $\{x_1, x_2, \ldots, x_n\}$
Nonlinear Dual SVMs

- Decide on a (valid) kernel function $K_\Phi$

- Find a maximal-margin separating hyperplane in the transformed space, $[\hat{w}_0^*, \hat{w}_0^*]$, by solving the QP:

  $$\min \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \bar{a}_i \bar{a}_j y_i y_j K_\Phi(\bar{x}_i, \bar{x}_j) - \sum_{i=1}^{n} \bar{a}_i$$

  subject to $\sum_{i=1}^{n} \bar{a}_i y_i = 0$

  $\bar{a}_i \geq 0 \ \forall \ i \in \{1, ..., n\}$

- Return the corresponding predictor in the original space:

  $$g(\bar{x}) = \text{sign} \left( \sum_{i: \bar{a}_i > 0} \bar{a}_i^* y_i K_\Phi(\bar{x}_i, \bar{x}) + \bar{w}_0^* \right)$$
Gaussian-RBF Kernel
Gaussian-RBF Kernel

Complex, non-linear margin with a changing width
You can very simply go to the Soft-margin SVM using a non-linear transformation which is key because you can’t know that the kernel you choose corresponds to a space where the training data is linearly separable.

Nonlinear Soft-Margin Dual SVMs

- Decide on a (valid) kernel function $K_{\Phi}$
- Find a maximal-margin separating hyperplane in the transformed space, $[\tilde{W}^*, \tilde{w}_0^*]$, by solving the QP:

$$\text{minimize} \quad \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K_{\Phi}(\tilde{x}_i, \tilde{x}_j) - \sum_{i=1}^{n} \alpha_i$$

subject to

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

$$0 \leq \alpha_i \leq C \forall i \in \{1, ..., n\}$$

- Return the corresponding predictor in the original space:

$$g(\tilde{x}) = \text{sign}\left(\sum_{i: \alpha_i > 0} \alpha_i y_i K_{\Phi}(\tilde{x}_i, \tilde{x}) + \tilde{w}_0^*\right)$$
is a tradeoff parameter