UMONS
Université de Mons

Support Vector Machines and One-Class Classification

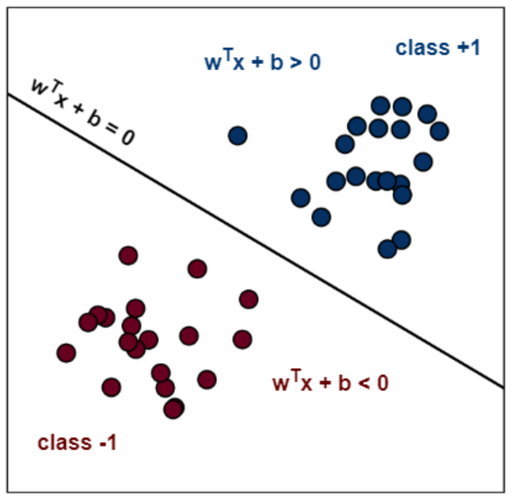
Sukanya Patra

March 17, 2023

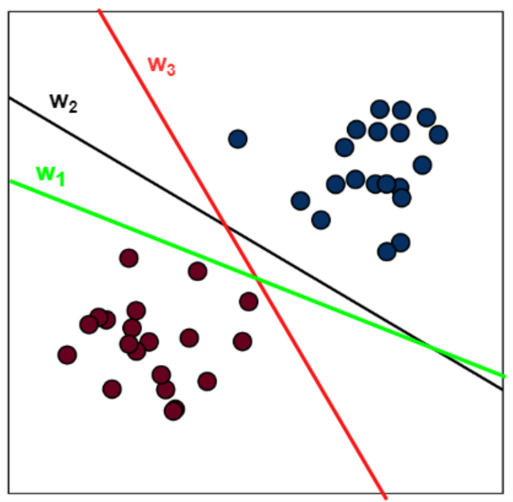
University of Mons

Contents

Linear Classifier

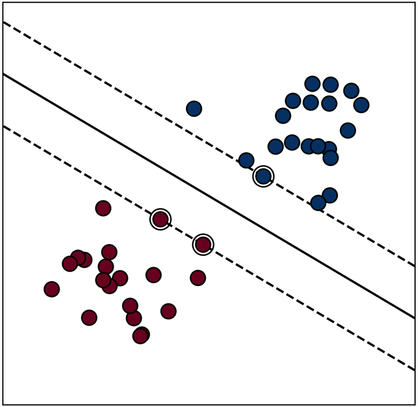


Which one is a good classifier?



Support Vector Machines (SVMs)

- SVMs choose the linear separator with the largest margin
- Proposed by Cortes et al. (1995).
- Good in terms of intuition, theory, practice
- Robust to outliers



Margin

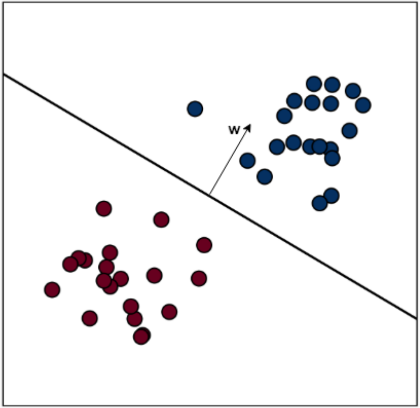
w is orthogonal to the hyperplane $w^T x + b = 0$.

Proof:

Consider two points $x^{(1)}$ and $x^{(2)}$ on the hyperplane. Thus,

$$w^T x^{(1)} + b = 0$$
$$w^T x^{(2)} + b = 0$$

So, $w^T(x^{(1)} - x^{(2)}) = 0$.



How to find the margin γ?

- $\frac{w}{||w||}$ is a unit vector along $\vec{w}$
- Suppose A represents point $x^{(i)}$
- B is given by $x^{(i)} - \gamma_i \frac{w}{||w||}$
- B lies on the decision boundary. Hence,

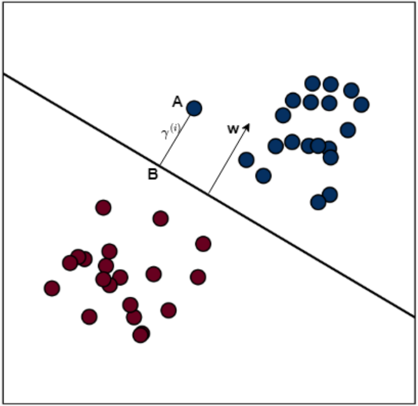
$$w^T(x^{(i)} - \gamma_i \frac{w}{||w||}) + b = 0$$

$$\gamma_i = \frac{w^T x^{(i)} + b}{||w||}$$

- For a training sample $(x^{(i)}, y^{(i)})$

$$\gamma_i = y^{(i)} \Big(\frac{w^T x^{(i)} + b}{||w||} \Big)$$

- The margin is $\gamma = \min_i \gamma_i$.



Optimal margin classifier

Now, we want to find a decision boundary that maximizes the distance to both classes

$$
\begin{aligned}
&\max_{\gamma, w, b} \quad \gamma \\
\text{such that} \quad & y^{(i)}\left(\frac{w^T x^{(i)} + b}{||w||}\right) \geq \gamma, \qquad i = 1, \cdots, n
\end{aligned}
$$

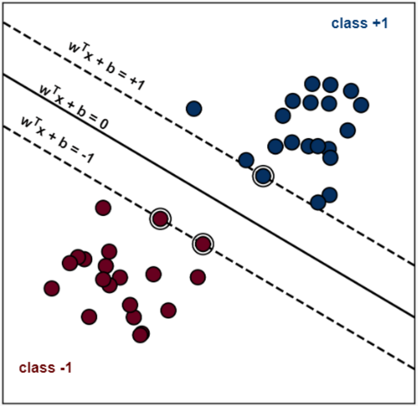
Optimal margin classifier (cont.)

- Maximizing the distance of all points to the decision boundary is the same as maximizing the distance to the closest points.
- The points closest to the decision boundary are called Support Vectors.
- For any plane, we can scale the w and b of the equation $w^T x + b = 0$ so that support vectors lie on the planes:

$$w^T x + b = \pm 1, \quad \text{depending on the class.}$$

Optimal margin classifier (cont.)

- The distance of the support vectors on planes $w^T x + b = \pm 1$, to the decision boundary is $\frac{1}{||w||}$
- Thus, we can define the margin as the distance to its support vectors, $\frac{2}{||w||}$.



Hard margin SVM

- We can reformulate the optimization problem as:

$$
\begin{aligned}
\max_{w,b} \quad & \frac{2}{||w||} \\
\text{such that} \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1, \quad i = 1, \cdots, n
\end{aligned}
$$

- Notice, maximizing $2/||w||$ is similar to minimizing a support vector regularization term $\frac{1}{2}||w||^2$.

$$
\begin{aligned}
\min_{w,b} \quad & \frac{1}{2}||w||^2 \\
\text{such that} \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1, \quad i = 1, \cdots, n
\end{aligned}
\tag{1}
$$

How to solve this optimization problem?

$$
\begin{aligned}
\min_{w,b} \quad & \frac{1}{2}||w||^2 \\
\text{such that} \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1, \quad i = 1, \cdots, n
\end{aligned}
$$

We can rewrite the constraint as:

$$-y^{(i)}(w^T x^{(i)} + b) + 1 \leq 0$$

The Lagrangian function is:

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i [y^{(i)}(w^T x^{(i)} + b) - 1] \tag{2}$$

How to solve this optimization problem? (cont.)

1. Minimize $\mathcal{L}(w, b, \alpha)$ with respect to w

$$\nabla_w \mathcal{L}(w, b, \alpha) = w - \sum_{i=1}^{n} \alpha_i y^{(i)} x^{(i)} = 0 \quad \Rightarrow w = \sum_{i=1}^{n} \alpha_i y^{(i)} x^{(i)}$$

2. Minimize $\mathcal{L}(w, b, \alpha)$ with respect to b

$$\nabla_b \mathcal{L}(w, b, \alpha) = \sum_{i=1}^{n} \alpha_i y^{(i)} = 0$$

3. Plug in the results in Equation 2:

$$\mathcal{L}(w, b, \alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1} y^{(i)} y^{(j)} \alpha_i \alpha_j (x^{(i)})^T x^{(j)}$$

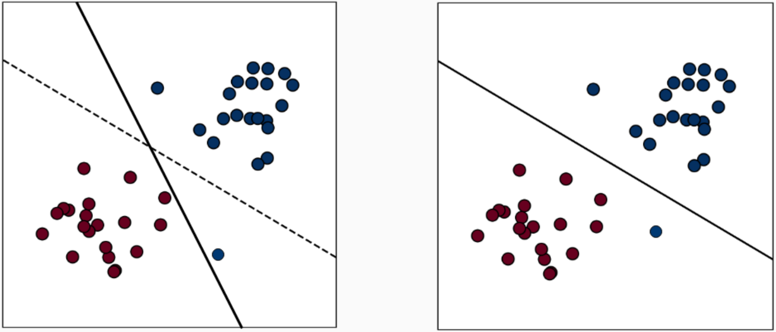
4. The dual becomes:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle \\ \text{such that} \quad & \alpha_i \geq 0, \quad i = 1, \cdots, n \\ & \sum_{i=1}^{n} \alpha_i y^{(i)} = 0 \end{aligned} \tag{3}$$

All the points $x^{(i)}$ with $\alpha_i > 0$ are support vectors.

Non-separable data

- Maximizing margin is fine as long as data is linearly separable.
- If the data contains outlier, performance might be sacrificed with very narrow margin.



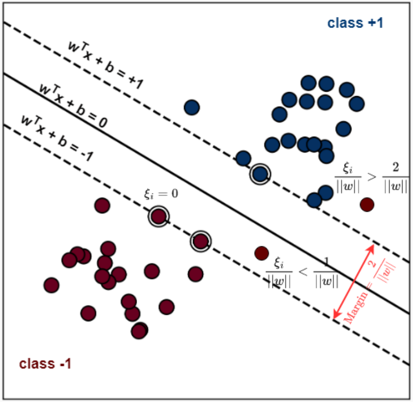
- There is a trade-off between maximizing margin and minimizing the error.

Slack variable

- A non-negative slack variable ξ_i is introduced for each data point $x^{(i)}$

- **Margin violation:** If a point lies on the correct side of boundary but is inside margin then $0 < \xi_i < 1$

- **Misclassification:** If a point lies on the wrong side of boundary $\xi_i > 1$



Soft margin SVM

- We can reformulate the optimization problem as follows:

$$\begin{aligned} \min_{w,b,\xi_i} \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i \\ \text{s. t.} \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i, \\ & \qquad\qquad i = 1, \cdots, n \\ & \xi_i \geq 0, \quad i = 1, \cdots, n \end{aligned}$$

- A hyperparameter $C > 0$ controls the trade-off between the slack variable penalty and the margin.
- Small C penalizes errors less and hence the classifier will have a large margin.
- Large C penalizes errors more and hence the classifier will accept narrow margins.
- Setting $C = \infty$ produces the hard margin solution.

How to solve soft margin optimization problem?

1. Form the Lagrangian:

$$\mathcal{L}(w,b,\xi,\alpha,r) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i[y^{(i)}(w^T x^{(i)} + b) - 1 + \xi_i] - \sum_{i=1}^{n} r_i\xi_i$$

2. Minimize $\mathcal{L}(w,b,\xi,\alpha,r)$ with respect to w

$$\nabla_w\mathcal{L}(w,b,\xi,\alpha,r) = w - \sum_{i=1}^{n}\alpha_i y^{(i)}x^{(i)} = 0 \quad \Rightarrow w = \sum_{i=1}^{n}\alpha_i y^{(i)}x^{(i)}$$

3. Minimize $\mathcal{L}(w,b,\xi,\alpha,r)$ with respect to b

$$\nabla_b\mathcal{L}(w,b,\xi,\alpha,r) = \sum_{i=1}^{n}\alpha_i y^{(i)} = 0$$

4. By plugging in the results, the dual problem becomes:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}y^{(i)}y^{(j)}\alpha_i\alpha_j\langle x^{(i)}, x^{(j)}\rangle \\ \text{such that} \quad & 0 \le \alpha_i \le C, \quad i = 1,\cdots,n \\ & \sum_{i=1}^{n}\alpha_i y^{(i)} = 0 \end{aligned} \tag{4}$$

All the points $x^{(i)}$ with $\alpha_i > 0$ are the support vectors.

Loss function

- Recall the optimization problem for soft margin SVM

$$\begin{aligned} \min_{w,b,\xi_i} \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{n} \xi_i \\ \text{s. t.} \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i, \quad i = 1, \cdots, n \\ & \xi_i \geq 0, \quad i = 1, \cdots, n \end{aligned}$$

- For a fixed pair of w and b,

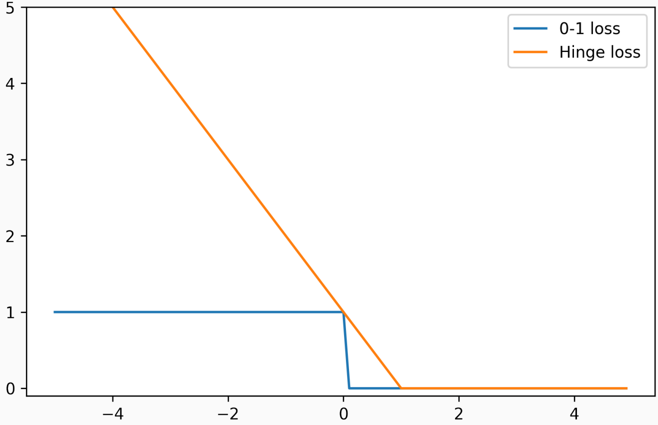
$$\xi_i = \begin{cases} 0 & \text{if } y^{(i)}(w^T x^{(i)} + b) \geq 1 \\ 1 - y^{(i)}(w^T x^{(i)} + b) & \text{if } y^{(i)}(w^T x^{(i)} + b) < 1 \end{cases}$$

$$\Rightarrow \xi_i = \max(0, 1 - y^{(i)}(w^T x^{(i)} + b))$$

- Thus, the learning problem becomes,

$$\min_{w,b} \quad \frac{1}{2}||w||^2 + C\sum_{i=1}^{n} \underbrace{\max(0, 1 - y^{(i)}(w^T x^{(i)} + b))}_{\text{loss function}}$$

Loss function (cont.)



- SVM uses hinge loss: $\max(0, 1 - y^{(i)}(w^T x^{(i)} + b))$
- Hinge loss upper bounds 0-1 loss: $\mathbb{I}[y^{(i)} \neq sign(w^T x^{(i)} + b)]$

Non-linear boundary

- Both hard margin and soft margin SVM has a linear decision boundary.
- Real-world datasets might not be linearly separable.
- **Solution**: Map the data into a feature space where it is linearly separable.
  - Apply transformation $\phi : \mathbb{R}^d \to \mathbb{R}^{d'}$ on the data

$$x^{(i)} \to \phi(x^{(i)})$$

  - Fit an SVM on the transformed data

$$\{\phi(x^{(1)}), \cdots, \phi(x^{(n)})\}$$

- In practice, $\mathbb{R}^{d'}$ is a very high dimensional space which makes computing ϕ for each sample computationally expensive.

Inner Product

- Recall the dual in the optimization problem of hard margin SVM

$$
\begin{aligned}
\max_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle \\
\text{such that} \quad & \alpha_i \geq 0, \quad i = 1, \cdots, n \\
& \sum_{i=1}^{n} \alpha_i y^{(i)} = 0
\end{aligned}
$$

- Note that the data is only being used for the inner product term $\langle x^{(i)}, x^{(j)} \rangle$ which captures the similarity between two vectors $x^{(i)}$ and $x^{(j)}$

- Thus we are interested in computing $\langle \phi(x^{(i)}), \phi(x^{(j)}) \rangle$ and not explicitly the terms $\phi(x^{(i)})$

Kernel

- Given a transformation function $\phi : \mathbb{R}^d \to \mathbb{R}^{d'}$, the kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is defined by

$$K(x^{(i)}, x^{(j)}) = \phi(x^{(i)}) \times \phi(x^{(j)}), \qquad x^{(i)}, x^{(j)} \in \mathbb{R}^d$$

- Thus, kernel function measure similarity of vectors without explicitly defining the transformation ϕ

- Given a choice of kernel function K, the dual becomes

$$\max_{\alpha} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1} y^{(i)} y^{(j)} \alpha_i \alpha_j K(x^{(i)}, x^{(j)})$$

- Therefore, kernel function allows training SVM in the feature space. Often referred to as kernel trick.

Example: Polynomial kernel

- Consider two vectors belonging to $\mathbb{R}^2$: $x = (x_1, x_2)$ and $y = (y_1, y_2)$
- Applying a polynomial transformation function $\phi : \mathbb{R}^2 \to \mathbb{R}^3$

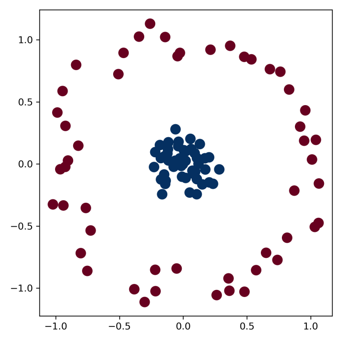
$$
\begin{aligned}
\phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2) \\
\phi(y) = (y_1^2, y_2^2, \sqrt{2}y_1y_2)
\end{aligned}
$$

- The inner product can be written as a kernel function K

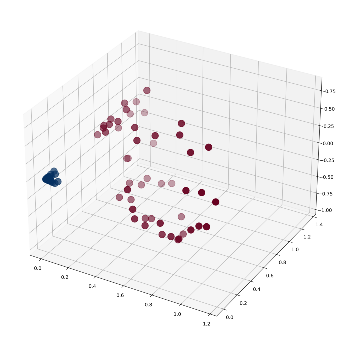
$$
\begin{aligned}
K(x, y) &= \phi(x)^T\phi(y) \\
&= x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2 \\
&= ((x_1, x_2)^T(y_1, y_2))^2 \\
&= (x^Ty)^2
\end{aligned}
$$

- Makes computing inner product computationally cheaper

Example: Polynomial kernel (cont.)



Original data



After the transformation

Commonly used kernel functions

- **Polynomial Kernel**

$$K(x^{(i)}, x^{(j)}) = ((x^{(i)})^T x^{(j)} + 1)^p,$$

where p is a hyperparameter

- **Radial Basis Function Kernel**

$$K(x^{(i)}, x^{(j)}) = exp\Big(-\frac{||x^{(i)} - x^{(j)}||^2}{2\sigma^2}\Big),$$

where σ is a hyperparameter

- **Sigmoid Kernel**

$$K(x^{(i)}, x^{(j)}) = tanh(\kappa (x^{(i)})^T x^{(j)} + \theta),$$

where κ and θ are a hyperparameters

Contents

One-class classification

One-class classification is a special type of classification dealing with a normal and a abnormal class

- *Normal* class is well sampled
- *Abnormal* class is sparsely sampled or completely absent

Example: Problem of machine diagnosis based on various sensor measurements

- Sampling measurements for a normally working machine is relatively cheap and easy
- Sampling measurements from faulty machine would require damaging the machine in various ways

One-Class SVMs

- Suppose, we only have data points from the positive class i.e., $y^{(i)} = 1$ for $i = 1, \cdots n$.
- The goal is to develop an algorithm which returns a function f that takes the value $+1$ in a "small" region capturing most of the data points, and -1 elsewhere.
- The strategy is to map the data into the feature space corresponding to the kernel, and to separate them from the origin with maximum margin.
- For a new point z, the value $f(z)$ is determined by evaluating which side of the hyperplane it falls on, in feature space.
- To separate data from the origin, we can maximize the distance ρ of the decision boundary from the origin.
- Proposed by Schölkopf et al. (2001).

One-Class SVM optimization problem

- The optimization problem can be formulated as:

$$\min_{w,\xi_i} \quad \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i - \rho$$

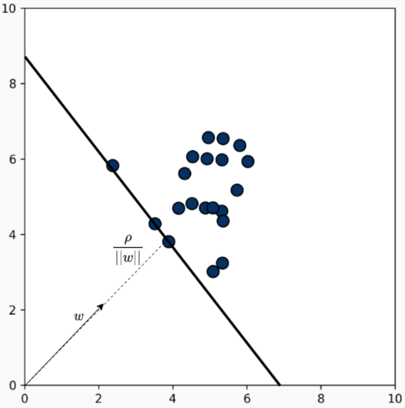
$$\text{s. t.} \quad w^T\Phi(x^{(i)}) \geq \rho - \xi_i, \quad i = 1, \cdots, n$$

$$\xi_i \geq 0, \quad i = 1, \cdots, n$$

- The decision function is

$$f(x^{(i)}) = sign((w^T\Phi(x^{(i)})) - \rho) \tag{5}$$

- The decision function is positive for most training examples $x^{(i)}$.



How to solve One-Class SVM optimization problem?

1. Form the Lagrangian: $\alpha_i, \beta_i \geq 0$

$$\mathcal{L}(w, \xi, \rho, \alpha, \beta) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i - \rho - \sum_{i=1}^{n}\alpha_i((w^T\Phi(x^{(i)})) - \rho + \xi_i) - \sum_{i=1}^{n}\beta_i\xi_i \quad (6)$$

2. Minimize $\mathcal{L}(w, \xi, \rho, \alpha, \beta)$ **with respect to** w

$$\nabla_w \mathcal{L}(w, \xi, \rho, \alpha, \beta) = w - \sum_{i=1}^{n}\alpha_i\Phi(x^{(i)}) = 0 \quad \Rightarrow w = \sum_{i=1}^{n}\alpha_i\Phi(x^{(i)}) \quad (7)$$

3. Minimize $\mathcal{L}(w, \xi, \rho, \alpha, \beta)$ **with respect to** ξ

$$\nabla_\xi \mathcal{L}(w, \xi, \rho, \alpha, \beta) = C - \alpha_i - \beta_i = 0 \quad \Rightarrow \alpha_i = C - \beta_i \leq C \quad (8)$$

In Equation 8, all the points $x^{(i)}$ with $\alpha_i > 0$ are called support vectors.

4. Minimize $\mathcal{L}(w, \xi, \rho, \alpha, \beta)$ **with respect to** ρ

$$\nabla_\rho \mathcal{L}(w, \xi, \rho, \alpha, \beta) = -1 + \sum_{i=1}^{n}\alpha_i = 0 \quad \Rightarrow \sum_{i=1}^{n}\alpha_i = 1 \quad (9)$$

5. By plugging in the results, the dual becomes:

$$\begin{aligned} &\min_{\alpha} \quad \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j k(x^{(i)}, x^{(j)}) \\ &\text{such that} \quad 0 \leq \alpha_i \leq C, \quad i = 1, \cdots, n, \quad \sum_{i=1}^{n}\alpha_i = 1 \end{aligned} \quad (10)$$

How to solve One-Class SVM optimization problem? (cont.)

- By substituting w from Equation 7 to Equation 5 we obtain,

$$F(x) = sign\Big(\sum_{i=1}^{n} \alpha_i k(x^{(i)}, x^{(j)}) - \rho\Big)$$

- The points which lie on the line $\sum_{i=1}^{n} \alpha_i k(x^{(i)}, x^{(j)}) - \rho = 0$ are the support vectors.
- Therefore, we can calculate ρ,

$$\rho = (w \cdot \Phi(x^{(i)})) = \sum_{i=1}^{n} \alpha_i k(x^{(i)}, x^{(j)})$$

Hyperparameters in One-Class SVM

- The trade-off parameter C can be defined as $C = \frac{1}{\nu n}$ where n is total number of available data points and $\nu \in (0, 1]$.

- For $\rho \neq 0$
  1. ν is an upper bound on the fraction of outliers.
  2. ν is a lower bound on the fraction of support vectors.

- if $\nu \to \infty$ the second inequality constraint in Equation 10 becomes void. Then the problem reduces to the hard margin algorithm.

Support Vector Data Description (SVDD)

- We can also use hypersphere to perform one class classification.
- Proposed by Tax and Duin (2004).
- We want to define a hypersphere with center c and radius R such that most of the training examples are inside the sphere,

$$||x^{(i)} - c||^2 \leq R^2$$

- Now, we want to make the hypershere as small as possible. Then the optimization function can be formulated as:

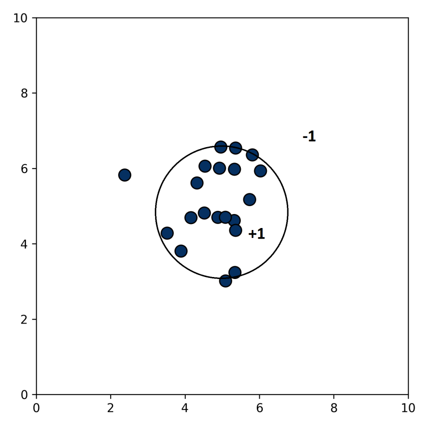
$$\begin{aligned} \min_{R,c} \quad & R^2 \\ \text{s. t.} \quad & ||x^{(i)} - c||^2 \leq R^2, \quad i \in \{1, \dots, n\} \end{aligned}$$

SVDD (cont.)

- To allow some outliers we can introduce some slack variable $\xi_i \geq 0$. Then the optimization problem becomes,

$$\begin{aligned} \min_{R,c,\xi_i} \quad & R^2 + C\sum_i \xi_i \\ \text{s. t.} \quad & ||x^{(i)} - c||^2 \leq R^2 + \xi_i, \\ & \xi_i \geq 0, \quad i \in \{1, \ldots, n\} \end{aligned}$$

- The parameter C controls the trade-off between the volume of the sphere and the errors.



How to solve SVDD optimization problem?

1. Form the Lagrangian: $\alpha_i, \beta_i \geq 0$

$$\mathcal{L}(R, c, \xi, \alpha, \beta) = R^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i[R^2 + \xi_i - ||x^{(i)} - c||^2] - \sum_{i=1}^{n}\beta_i\xi_i \tag{11}$$

2. Minimize $\mathcal{L}$ **with respect to** R

$$\nabla_R\mathcal{L} = 2R - \sum_{i=1}^{n} 2R\alpha_i = 0 \quad \Rightarrow \sum_{i=1}^{n}\alpha_i = 1$$

3. Minimize $\mathcal{L}$ **with respect to** ξ

$$\begin{aligned} \nabla_\xi\mathcal{L} &= C - \alpha_i - \beta_i = 0 \\ &\Rightarrow \alpha_i = C - \beta_i \\ &\Rightarrow 0 \leq \alpha_i \leq C, \quad \text{as } \alpha_i \geq 0 \text{ and } \beta_i \geq 0 \end{aligned}$$

   All the points $x^{(i)}$ with $\alpha_i > 0$ are called support vectors.

4. Minimize $\mathcal{L}$ **with respect to** c

$$\nabla_c\mathcal{L} = \sum_{i=1}^{n}\alpha_i c - \sum_{i=1}^{n}\alpha_i x^{(i)} = 0 \quad \Rightarrow c = \frac{\sum_{i=1}^{n}\alpha_i x^{(i)}}{\sum_{i=1}^{n}\alpha_i}$$

5. By plugging in the results, the dual becomes:

$$\max_{\alpha} \quad \sum_{i=1}^{n}\alpha_i(x^{(i)} \cdot x^{(i)}) - \sum_{i,j=1}^{n}\alpha_i\alpha_j(x^{(i)}, x^{(j)}) \quad \text{s. t.} \quad 0 \leq \alpha_i \leq C$$

Support vectors in SVDD

- For a point $x^{(i)}$, three scenarios can arise:
  - $||x^{(i)} - c||^2 < R^2 \rightarrow \alpha_i = 0, \beta_i = 0$
  - $||x^{(i)} - c||^2 = R^2 \rightarrow 0 < \alpha_i < C, \beta_i = 0$
  - $||x^{(i)} - c||^2 > R^2 \rightarrow \alpha_i = C, \beta_i > 0$

- The points $x^{(i)}$ with $\alpha_i > 0$ are support vectors.

- A test point z is assigned positive label when

$$||z - c||^2 = (z \cdot z) - 2\sum_{i=1}^{n} \alpha_i (z \cdot x^{(i)}) + \sum_{i,j=1}^{n} \alpha_i \alpha_j x^{(i)} x^{(j)} \leq R^2$$

Radius of the hypersphere

- R^2 is the distance from the center c to any support vector on the decision boundary.
- The support vectors which falls outside the decision boundary ($\alpha_i = C$) are not considered. Thus,

$$R^2 = (x^{(k)} \cdot x^{(k)}) - 2\sum_{i=1}^{n} \alpha_i (x^{(i)} \cdot x^{(k)}) + \sum_{i,j=1}^{n} \alpha_i \alpha_j x^{(i)} x^{(j)},$$

where $x^{(k)}$ is a support vector for which $\alpha_k < C$

Summary

- SVMs maximize the margin along with learning the decision boundary.
- The decision boundary learned by SVMs depends only on the support vectors.
- Soft margin SVMs are robust to outliers.
- The kernel trick allows us to learn the decision boundary of non-linearly separable data.
- One-class SVM is used for tasks such as anomaly detection and outlier detection.
- One-class SVMs use a hyperplane to describe the data in feature space, whereas SVDDs use a hypersphere.

References

Cortes, C., Vapnik, V., and Saitta, L. (1995). Support-vector networks. *Machine Learning 1995 20:3*, 20(3):273–297.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (2001). Estimating the Support of a High-Dimensional Distribution. *Neural Computation*, 13(7):1443–1471.

Tax, D. M. and Duin, R. P. (2004). Support Vector Data Description. *Machine Learning 2004 54:1*, 54(1):45–66.

Discussion

- Do you see any potential challenges of using SVM?
-