Support Vector Machines

Classification with SVM, kernel trick

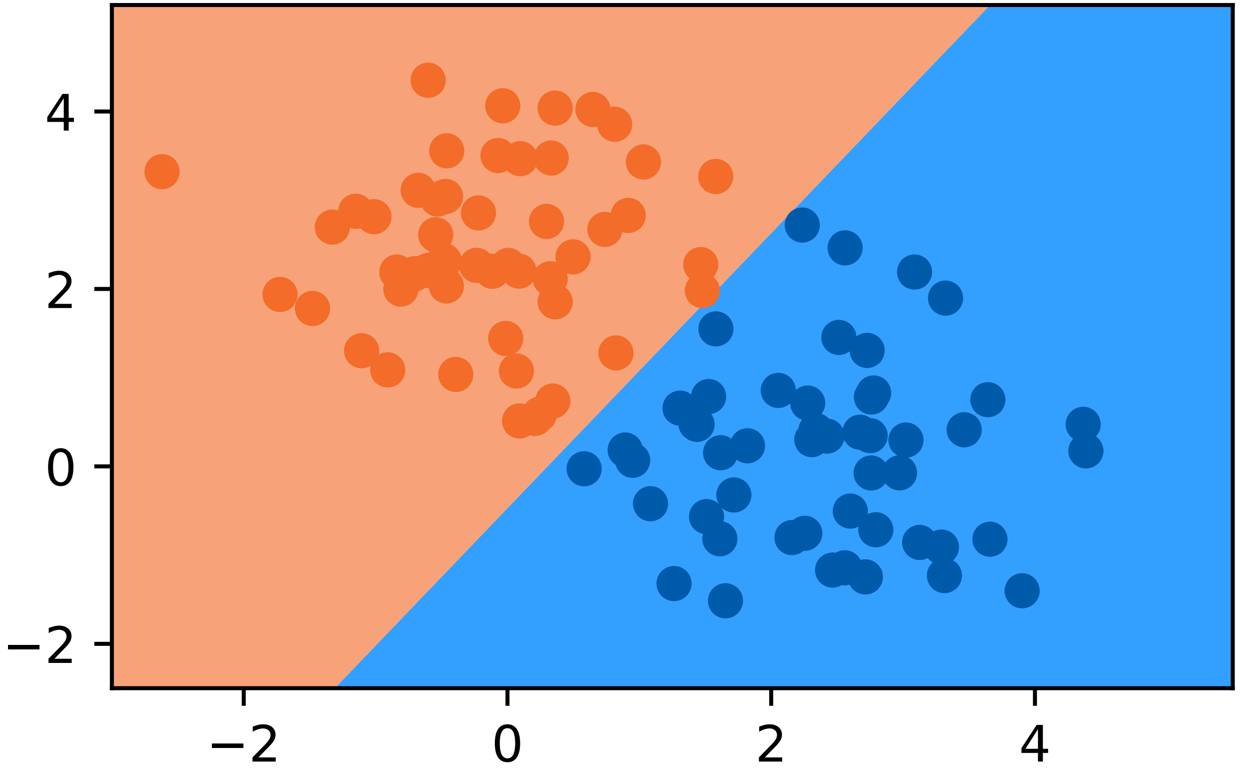
Machine Learning and Data Mining, 2021

Artem Maevskiy
National Research University Higher School of Economics

LAMBDA • HSE

General Idea
(linearly separable case)

Classification with linear models



$$\hat{f}(x) = \text{sign}[w^{\mathrm{T}}x + w_0]$$

$$y \in \{-1, 1\}$$

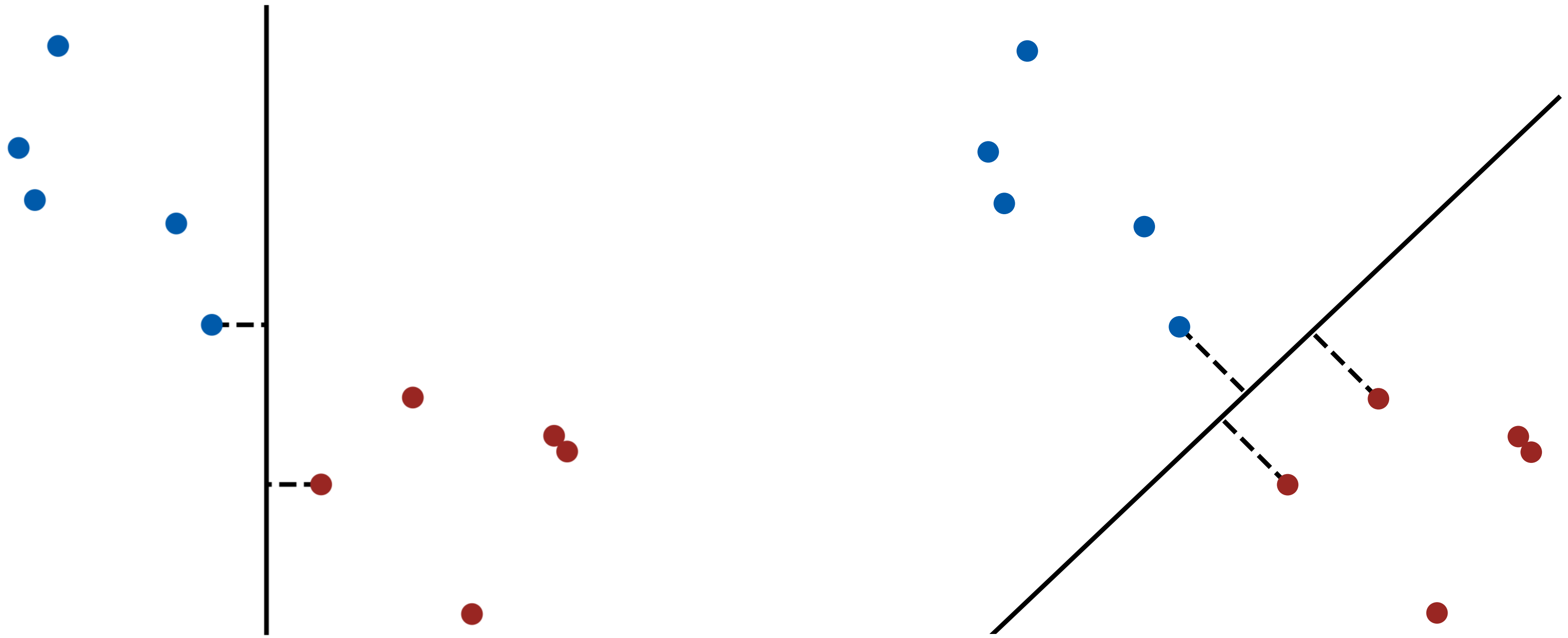
Separating hyperplane:

$$w^{\mathrm{T}}x + w_0 = 0$$

Optimal hyperplane

Assume a separating hyperplane exists (task is linearly separable)

Idea: find the best hyperplane by maximizing the distance to the closest data points



Mathematical formulation

Correct classification if:

$$\begin{cases} w^{\mathrm{T}}x + w_0 > 0, & y = +1 \\ w^{\mathrm{T}}x + w_0 < 0, & y = -1 \end{cases}$$

or equivalently:

$$y\left(w^{\mathrm{T}}x + w_0\right) > 0$$

Mathematical formulation

Correct classification if:

$$\begin{cases} w^{\mathrm{T}}x + w_0 > 0, & y = +1 \\ w^{\mathrm{T}}x + w_0 < 0, & y = -1 \end{cases}$$

or equivalently:

$$y\left(w^{\mathrm{T}}x + w_0\right) > 0$$

defined up to a multiplicative constant
We can choose this constant s.t. for the closest point: $y_{\text{closest}}\left(w^{T}x_{\text{closest}} + w_0\right) = 1$

Mathematical formulation

Correct classification if:

$$\begin{cases} w^{\mathrm{T}}x + w_0 > 0, & y = +1 \\ w^{\mathrm{T}}x + w_0 < 0, & y = -1 \end{cases}$$

or equivalently:

$$y\left(w^{\mathrm{T}}x + w_0\right) > 0$$

defined up to a multiplicative constant
We can choose this constant s.t. for the closest point: $\boldsymbol{y_{\text{closest}}\left(w^T x_{\text{closest}} + w_0\right) = 1}$

So for all points:

$$y\left(w^{\mathrm{T}}x + w_0\right) \geq 1$$

Mathematical formulation

Correct classification if:

$$\begin{cases} w^{\mathrm{T}}x + w_0 > 0, & y = +1 \\ w^{\mathrm{T}}x + w_0 < 0, & y = -1 \end{cases}$$

or equivalently:

$$y\left(w^{\mathrm{T}}x + w_0\right) > 0$$

defined up to a multiplicative constant
We can choose this constant s.t. for the closest point: $y_{\text{closest}}\left(w^T x_{\text{closest}} + w_0\right) = 1$

So for all points:

$$y\left(w^{\mathrm{T}}x + w_0\right) \geq 1$$

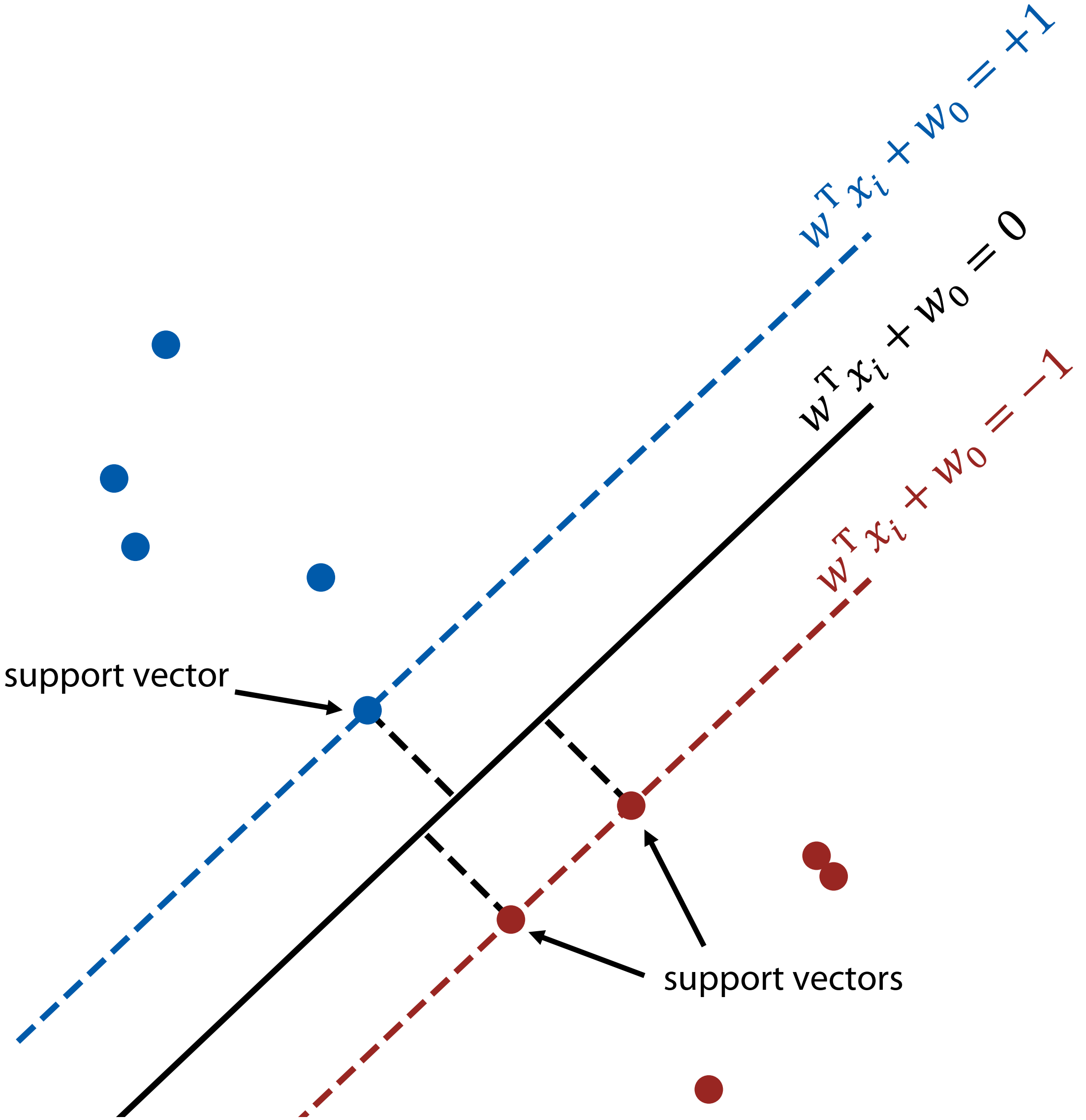
Distance to the closest point is:

$$h = y_{\text{closest}} \frac{\left(w^{\mathrm{T}}x_{\text{closest}} + w_0\right)}{\|w\|} = \frac{1}{\|w\|}$$

Mathematical formulation

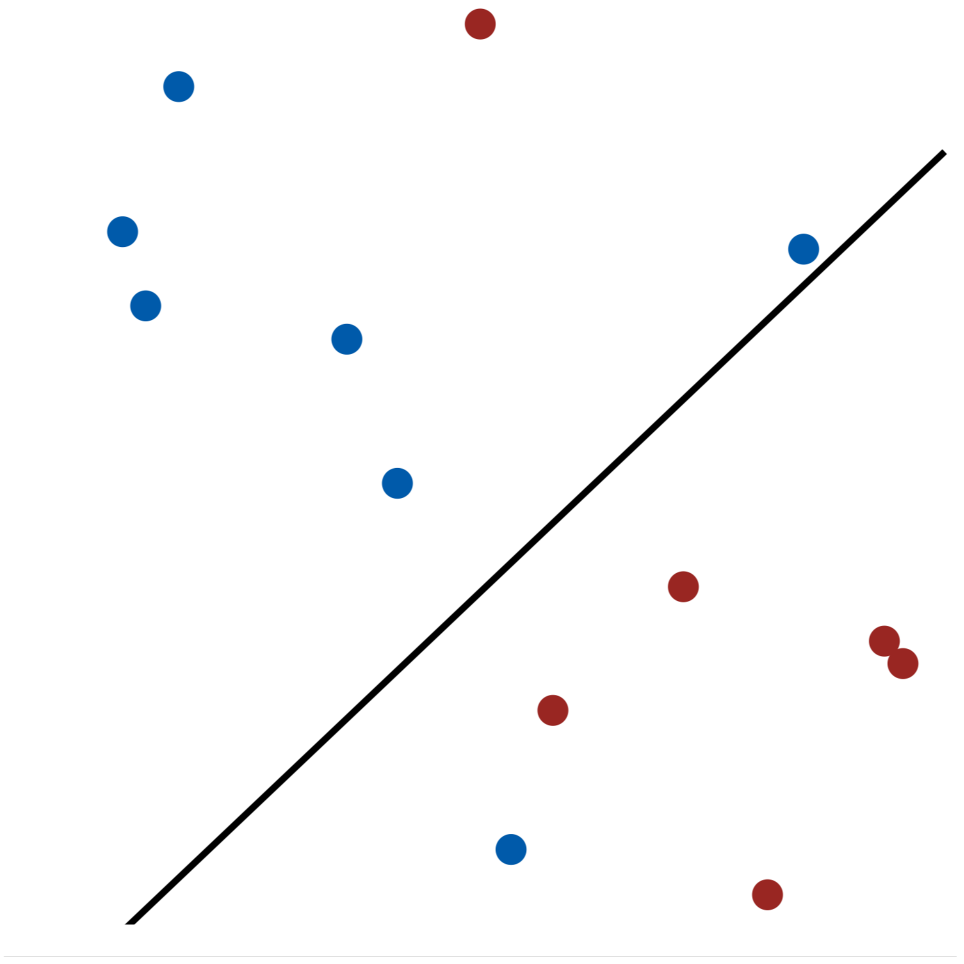
So the problem can be defined as:

$$\begin{cases} \dfrac{1}{2}\|w\|^2 \to \min\limits_{w,w_0} \\ y_i\left(w^{\mathrm{T}}x_i + w_0\right) \geq 1, \qquad i = 1, \dots, N \end{cases}$$



Nonseparable case

Slack variables

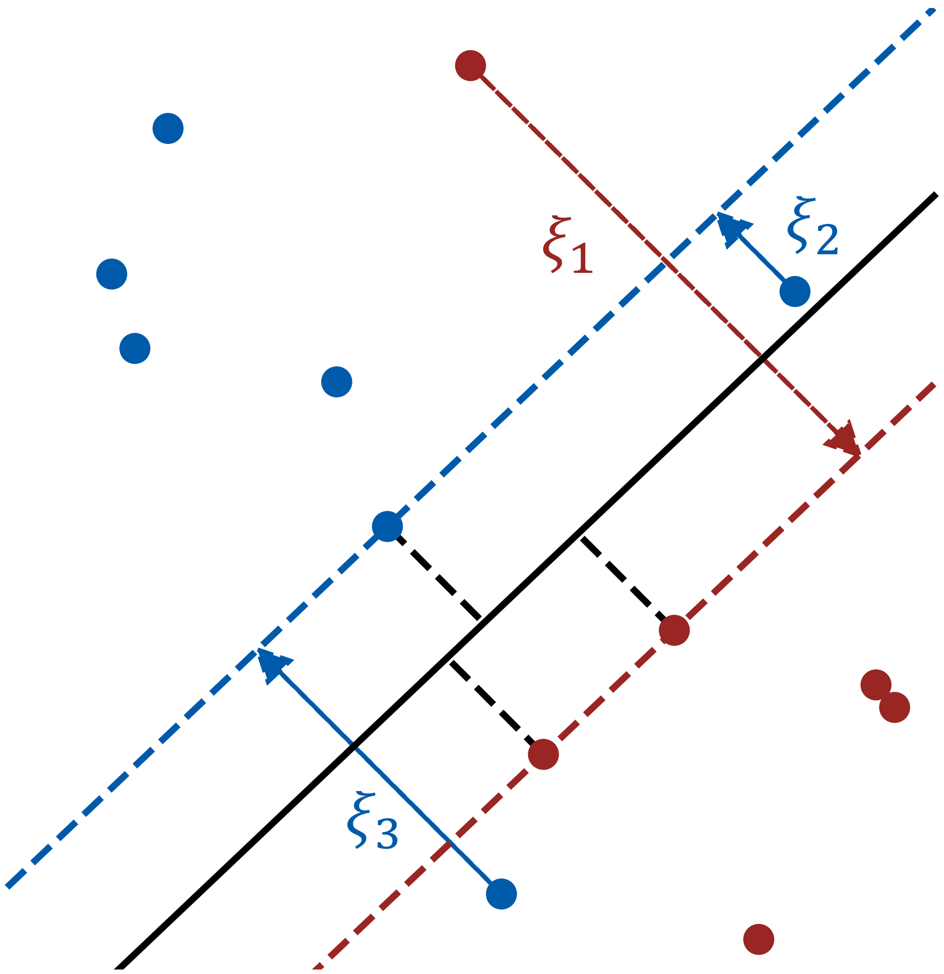


For nonseparable case, these conditions:

$$y_i\left(w^{\mathrm{T}}x_i + w_0\right) \geq 1, \qquad i = 1, \dots, N$$

cannot be satisfied simultaneously.

Slack variables



For nonseparable case, these conditions:

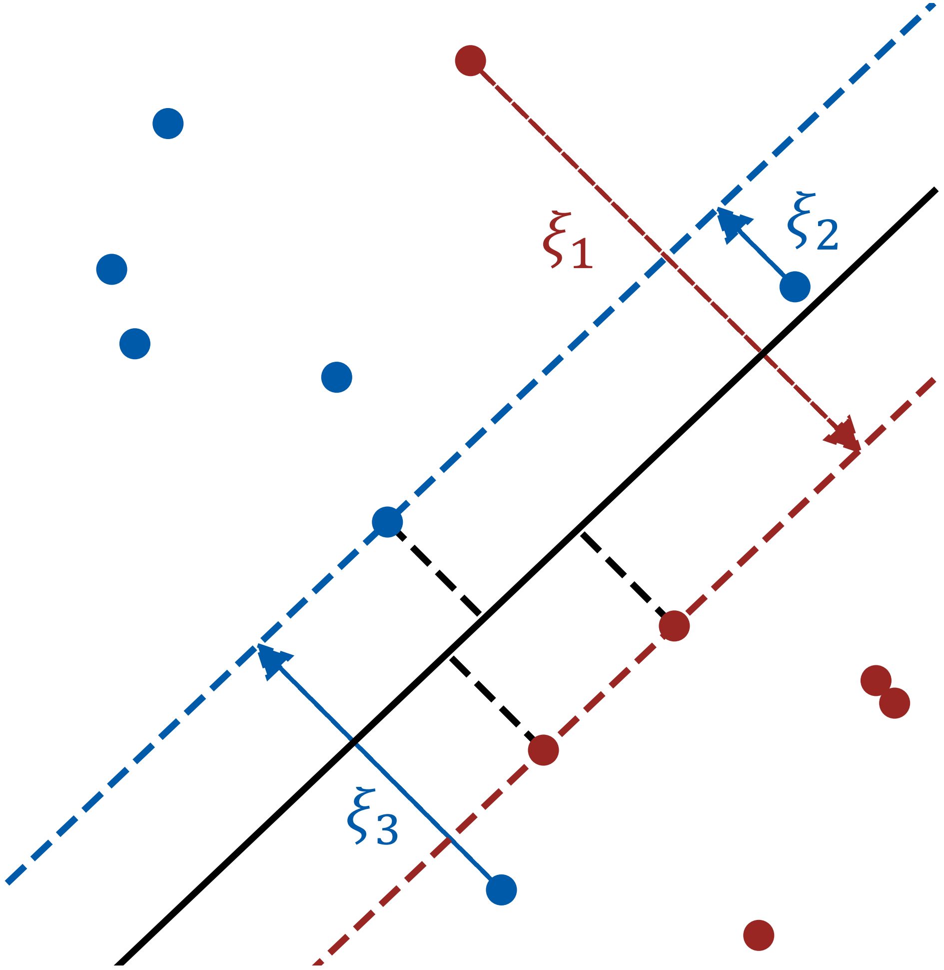
$$y_i(w^{\mathrm{T}}x_i + w_0) \geq 1, \qquad i = 1, \dots, N$$

cannot be satisfied simultaneously.

Need to introduce slack variables ξ_i:

$$y_i(w^{\mathrm{T}}x_i + w_0) \geq 1 - \xi_i,$$
$$\xi_i \geq 0, \qquad i = 1, \dots, N$$

Slack variables



For nonseparable case, these conditions:

$$y_i\left(w^{\mathrm{T}} x_i + w_0\right) \geq 1, \qquad i = 1, \dots, N$$

cannot be satisfied simultaneously.

Need to introduce slack variables ξ_i:

$$y_i\left(w^{\mathrm{T}} x_i + w_0\right) \geq 1 - \xi_i,$$
$$\xi_i \geq 0, \qquad i = 1, \dots, N$$

And the objective function becomes:

$$\frac{1}{2}\|w\|^2 + C \sum_{i=1}^{N} \xi_i \to \min_{w, w_0, \xi}$$

Solution

To solve:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \rightarrow \min_{w,w_0,\xi}$$

subject to:

$$y_i\left(w^{\mathrm{T}}x_i + w_0\right) \geq 1 - \xi_i,$$
$$\xi_i \geq 0, \qquad i = 1, \dots, N$$

Solution

To solve:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \to \min_{w,w_0,\xi}$$

subject to:

$$y_i\left(w^{\mathrm{T}}x_i + w_0\right) \geq 1 - \xi_i,$$
$$\xi_i \geq 0, \qquad i = 1, \dots, N$$

define the Lagrangian:

$$L(w, w_0, \xi, \alpha, r) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i\left(y_i\left(w^{\mathrm{T}}x_i + w_0\right) - 1 + \xi_i\right) - \sum_{i=1}^{N} r_i\xi_i$$

Solution

To solve:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \to \min_{w,w_0,\xi}$$

subject to:

$$y_i\left(w^{\mathrm{T}}x_i + w_0\right) \geq 1 - \xi_i,$$
$$\xi_i \geq 0, \qquad i = 1, \dots, N$$

define the Lagrangian:

$$L(w, w_0, \xi, \alpha, r) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i\left(y_i\left(w^{\mathrm{T}}x_i + w_0\right) - 1 + \xi_i\right) - \sum_{i=1}^{N}r_i\xi_i$$

Solution determined by the Karush–Kuhn–Tucker conditions:

$$\frac{\partial L}{\partial(w, w_0, \xi_i)} = 0 \qquad L \to \max_{\alpha,r}$$
$$\alpha_i \geq 0, \qquad r_i \geq 0$$
$$y_i\left(w^{\mathrm{T}}x_i + w_0\right) \geq 1 - \xi_i$$
$$\xi_i \geq 0,$$
$$\alpha_i\left(y_i\left(w^{\mathrm{T}}x_i + w_0\right) - 1 + \xi_i\right) = 0$$
$$r_i\xi_i = 0 \qquad i = 1, \dots, N$$

Solution

$$L(w, w_0, \xi, \alpha, r) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i\left(y_i\left(w^{\mathrm{T}}x_i + w_0\right) - 1 + \xi_i\right) - \sum_{i=1}^{N} r_i\xi_i$$

$$\frac{\partial L}{\partial w} = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{N}\alpha_i y_i x_i$$

i.e. the solution is a **linear combination** of the training objects.

Solution

$$L(w, w_0, \xi, \alpha, r) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i\left(y_i\left(w^{\mathrm{T}}x_i + w_0\right) - 1 + \xi_i\right) - \sum_{i=1}^{N} r_i\xi_i$$

$$\frac{\partial L}{\partial w} = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{N}\alpha_i y_i x_i$$

i.e. the solution is a **linear combination** of the training objects.

$$\frac{\partial L}{\partial w_0} = 0 \quad \Rightarrow \quad \sum_{i=1}^{N}\alpha_i y_i = 0$$

i.e. the combination coefficients need to be **balanced** between classes.

Solution

$$L(w, w_0, \xi, \alpha, r) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i\left(y_i\left(w^{\mathrm{T}}x_i + w_0\right) - 1 + \xi_i\right) - \sum_{i=1}^{N} r_i\xi_i$$

$$\frac{\partial L}{\partial w} = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{N}\alpha_i y_i x_i$$

i.e. the solution is a **linear combination** of the training objects.

$$\frac{\partial L}{\partial w_0} = 0 \quad \Rightarrow \quad \sum_{i=1}^{N}\alpha_i y_i = 0$$

i.e. the combination coefficients need to be **balanced** between classes.

$$\frac{\partial L}{\partial \xi_i} = 0 \quad \Rightarrow \quad C - \alpha_i - r_i = 0$$

Dual problem

Substituting these into the Lagrangian:

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i \,, \qquad \sum_{i=1}^{N} \alpha_i y_i = 0, \qquad C - \alpha_i - r_i = 0$$

we obtain the **dual problem**:

$$\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^{\mathrm{T}} x_j \to \max_{\alpha}$$

subject to:

$$\sum_{i=1}^{N} \alpha_i y_i = 0, \qquad 0 \leq \alpha_i \leq C$$

Support vectors

Due to these KKT conditions:

$$\alpha_i\left(y_i\left(w^{\mathrm{T}}x_i + w_0\right) - 1 + \xi_i\right) = 0, \qquad r_i\xi_i = 0$$

there are the following options for the training objects:

Support vectors

Due to these KKT conditions:

$$\alpha_i\left(y_i\left(w^{\mathrm{T}}x_i + w_0\right) - 1 + \xi_i\right) = 0, \qquad\qquad r_i\xi_i = 0$$

there are the following options for the training objects:

$$y_i\left(w^{\mathrm{T}}x_i + w_0\right) > 1 \;\;\Rightarrow\;\; \alpha_i = 0$$ **(non-informative vector)**

Support vectors

Due to these KKT conditions:

$$\alpha_i\left(y_i\left(w^{\mathrm{T}}x_i + w_0\right) - 1 + \xi_i\right) = 0, \qquad r_i\xi_i = 0$$

there are the following options for the training objects:

$$y_i\left(w^{\mathrm{T}}x_i + w_0\right) > 1 \;\Rightarrow\; \alpha_i = 0$$ **(non-informative vector)**

$$y_i\left(w^{\mathrm{T}}x_i + w_0\right) < 1 \;\Rightarrow\; \xi_i > 0,\; r_i = 0,\; \alpha_i = C$$ **(non-boundary support vector)**

Support vectors

Due to these KKT conditions:

$$\alpha_i\left(y_i\left(w^{\mathrm{T}}x_i + w_0\right) - 1 + \xi_i\right) = 0, \qquad\qquad r_i\xi_i = 0$$

there are the following options for the training objects:

$$y_i\left(w^{\mathrm{T}}x_i + w_0\right) > 1 \;\Rightarrow\; \alpha_i = 0$$ **(non-informative vector)**

$$y_i\left(w^{\mathrm{T}}x_i + w_0\right) < 1 \;\Rightarrow\; \xi_i > 0,\; r_i = 0,\; \alpha_i = C$$ **(non-boundary support vector)**

$$y_i\left(w^{\mathrm{T}}x_i + w_0\right) = 1 \;\Rightarrow\; \xi_i = 0,\; \alpha_i \in [0, C]$$ **(boundary support vector)**

Whole pipeline:

Solve the dual problem to find the optimal α^*

$$\sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^{\mathrm{T}} x_j \to \max_\alpha$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0, \qquad 0 \le \alpha_i \le C$$

Can be obtained from e.g. boundary support vectors from: $y_i\left(w^{\mathrm{T}} x_i + w_0\right) = 1$

Make predictions for new data:

$$\hat{y} = \mathrm{sign}\left(\sum_{i \in SV} \alpha_i^* y_i x_i^{\mathrm{T}} x + w_0\right)$$

Kernel trick

Whole pipeline:

Note that the dual problem and prediction depend on the data only through **scalar products**:

$$\sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^{\mathrm{T}} x_j \to \max_\alpha$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0, \qquad 0 \le \alpha_i \le C$$

$$\hat{y} = \operatorname{sign}\left(\sum_{i \in SV} \alpha_i^* y_i x_i^{\mathrm{T}} x + w_0\right)$$

Feature expansion

Suppose we want to expand our features:

$$x_i \to \phi(x_i)$$

Then, our solution would only depend on the scalar products: $\phi^{\mathrm{T}}(x_i)\phi(x_j)$.

Feature expansion

Suppose we want to expand our features:

$$x_i \to \phi(x_i)$$

Then, our solution would only depend on the scalar products: $\phi^{\mathrm{T}}(x_i)\phi(x_j)$.

E.g. for the following polynomial expansion: $(x_1, \mathrm{x}_2)_\mathrm{i} \to \left(\frac{1}{2}, x_1, \mathrm{x}_2, x_1^2, \sqrt{2}x_1\mathrm{x}_2, \mathrm{x}_2^2\right)_i$,

the scalar product equals to:

$$\frac{1}{4} + x_{1i}x_{1j} + x_{2i}x_{2j} + \left(x_{1i}x_{1j}\right)^2 + 2\left(x_{1i}x_{1j}\right)\left(x_{2i}x_{2j}\right) + \left(x_{2i}x_{2j}\right)^2 =$$

$$= \frac{1}{4} + x_i^{\mathrm{T}}x_j + \left(x_i^{\mathrm{T}}x_j\right)^2 = \left(x_i^{\mathrm{T}}x_j + \frac{1}{2}\right)^2$$

Feature expansion

Suppose we want to expand our features:

$$x_i \to \phi(x_i)$$

Then, our solution would only depend on the scalar products: $\phi^{\mathrm{T}}(x_i)\phi(x_j)$.

E.g. for the following polynomial expansion: $(x_1, \mathrm{x}_2)_{\mathrm{i}} \to \left(\frac{1}{2}, x_1, \mathrm{x}_2, x_1^2, \sqrt{2}x_1\mathrm{x}_2, \mathrm{x}_2^2\right)_i$,

the scalar product equals to:

$$\frac{1}{4} + x_{1i}x_{1j} + x_{2i}x_{2j} + \left(x_{1i}x_{1j}\right)^2 + 2\left(x_{1i}x_{1j}\right)\left(x_{2i}x_{2j}\right) + \left(x_{2i}x_{2j}\right)^2 =$$

$$= \frac{1}{4} + x_i^{\mathrm{T}}x_j + \left(x_i^{\mathrm{T}}x_j\right)^2 = \left(x_i^{\mathrm{T}}x_j + \frac{1}{2}\right)^2$$

So instead of doing the expansion we can replace all scalar products with:

$$x_i^{\mathrm{T}}x_j \to K\left(x_i, x_j\right) = \left(x_i^{\mathrm{T}}x_j + \frac{1}{2}\right)^2$$

RBF kernel

This trick allows for expansions that would normally be infeasible to compute, e.g. expansions to infinite dimension spaces.

Example: **Radial Basis Function** (RBF) kernel:

$$K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}$$

This kernel has maximum of 1 for $x_i = x_j$, and decays to 0 as the vectors become further apart. Hence, the solution averages the labels for nearby support vectors:

$$\hat{y} = \operatorname{sign}\left(\sum_{i \in SV} \alpha_i^* y_i K(x_i, x\,) + w_0\right)$$

Can any function be a kernel?

Note that the quadratic form of the dual problem is defined by the symmetric, positive semi-definite matrix XX^{T}:

$$\sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^{\mathrm{T}} x_j \to \max_{\alpha}$$

In fact, kernel functions should also have these properties (Mercer theorem):

Symmetry:

$$K(x_i, x_j) = K(x_j, x_i)$$

For every set $x_1, \dots, x_M$ the Gram matrix is positive semi-definite:

$$K(x_i, x_j) \equiv K_{ij} \quad \text{– p.s.d.}$$

Summary

- SVM is a linear classification model. In the linearly separable case it maximizes the distance from the closest training points to the decision boundary

Summary

- SVM is a linear classification model. In the linearly separable case it maximizes the distance from the closest training points to the decision boundary
- For the inseparable case, it balances between:
  - minimizing $\|w\|^2$, i.e. pushing the $w^{\mathrm{T}}x_i + w_0 = \pm 1$ lines further away from the decision boundary
  - maximizing the sum of margins for objects with $y_i\left(w^{\mathrm{T}}x_i + w_0\right) < 1$ (through slack variables)
- The balance is controlled by the constant C

Summary

- SVM is a linear classification model. In the linearly separable case it maximizes the distance from the closest training points to the decision boundary
- For the inseparable case, it balances between:
  - minimizing $\|w\|^2$, i.e. pushing the $w^{\mathrm{T}}x_i + w_0 = \pm 1$ lines further away from the decision boundary
  - maximizing the sum of margins for objects with $y_i\left(w^{\mathrm{T}}x_i + w_0\right) < 1$ (through slack variables)
- The balance is controlled by the constant C
- Solution only depends on the support vectors
  - Not robust to outliers as they always become support vectors

Summary

- SVM is a linear classification model. In the linearly separable case it maximizes the distance from the closest training points to the decision boundary
- For the inseparable case, it balances between:
  - minimizing $\|w\|^2$, i.e. pushing the $w^{\mathrm{T}}x_i + w_0 = \pm 1$ lines further away from the decision boundary
  - maximizing the sum of margins for objects with $y_i\left(w^{\mathrm{T}}x_i + w_0\right) < 1$ (through slack variables)
- The balance is controlled by the constant C
- Solution only depends on the support vectors
  - Not robust to outliers as they always become support vectors
- Kernel trick allows to expand features by just redefining the scalar product in the original feature space (i.e. almost no computational overhead)
  - This allows for infinite dimension representations
  - Can define kernels (similarity measures) for complex objects like strings, sets, graphs, etc.

Thank you!

amaevskij@hse.ru

SiLiKhon

hse_lambda

Artem Maevskiy