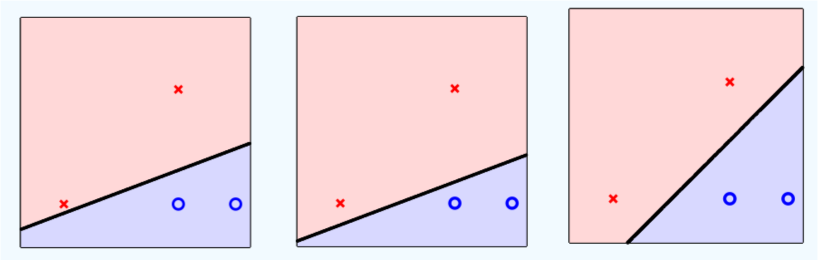
SVM

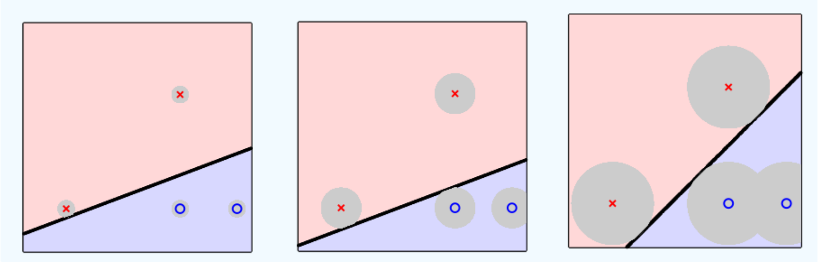
Given a linearly separable D, a linear decision boundary separating negatives from positives can be obtained using, for instance, **PLA** or **logistic regression**



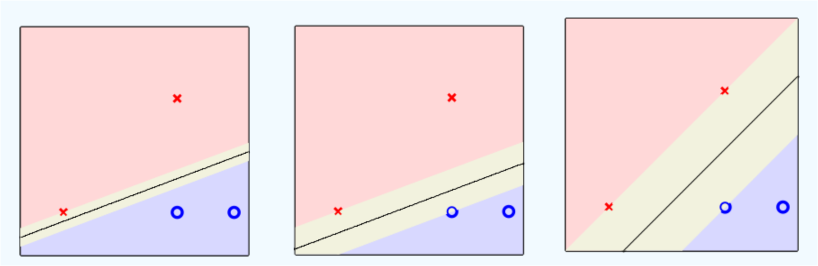
Is there one that is preferable than others ?

Intuition

Depending on where the separating line is, it is more or less robust to noise



Maximum margin



Any of these lines separate the negatives from the positives
They have margins of different sizes

Problem formulation

How to find the **hyperplane that maximizes the margin** ?

In **SVM**, this is achieved by formulating the problem as a quadratic programmin (QP) optimization problem

QP: optimization of quadratic functions with linear constraints on the variables

The problem we want to solve

$$\begin{aligned} \underset{\mathbf{w}}{\text{maximize}} \quad & \frac{1}{||\mathbf{w}||} \\ \text{subject to} \quad & \min_{n=1,\ldots,N} y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1 \end{aligned}$$

The constraint $\min_{n=1,\ldots,N} y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1$ implies $y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1$ which has the effect of forcing all examples to be classified correctly

The equality $\min_{n=1,\ldots,N} y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1$ implies that the distance of the closest point to the hyperplane is $\frac{1}{||\mathbf{w}||}$ (a nice objective function!)

The problem we want to solve

$$\begin{aligned} \underset{\mathbf{w}}{\text{maximize}} \quad & \frac{1}{||\mathbf{w}||} \\ \text{subject to} \quad & \min_{n=1,\dots,N} y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1 \end{aligned}$$

Equivalent formulation

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} \\ \text{subject to} \quad & \min_{n=1,\dots,N} y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1 \end{aligned}$$

Relaxed formulation

Original minimization formulation:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} \\ \text{subject to} \quad & \min_{n=1,\ldots,N} y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1 \end{aligned}$$

Equivalent relaxed formulation (this is a QP optimization):

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} \\ \text{subject to} \quad & y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1, n = 1, \ldots, N \end{aligned}$$

The equivalence can be proved by contradiction (see Chapter on SVM, page 7)

Solving a toy example by hand

$$X = \begin{bmatrix} 0 & 0 \\ 2 & 2 \\ 2 & 0 \\ 3 & 0 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix}$$

Contraints:

$$\begin{aligned} -b &\geq 1 \quad (1) \\ -(2w_1 + 2w_2 + b) &\geq 1 \quad (2) \\ 2w_1 + b &\geq 1 \quad (3) \\ 3w_1 + b &\geq 1 \quad (4) \end{aligned}$$

Solving a toy example by hand

$$\begin{aligned} -b &\geq 1 \quad (1)\\ -(2w_1 + 2w_2 + b) &\geq 1 \quad (2)\\ 2w_1 + b &\geq 1 \quad (3)\\ 3w_1 + b &\geq 1 \quad (4) \end{aligned}$$

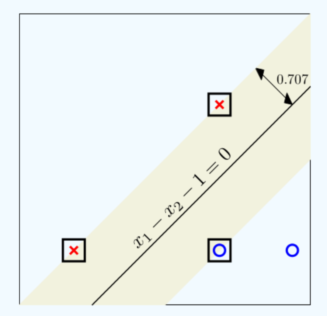
- From (3) and (1)
  $2w_1 + b \geq 1 \rightsquigarrow 2w_1 \geq 1 - b \rightsquigarrow w_1 \geq \frac{1}{2}(1-b)$ && $b \leq -1$
  $\Longrightarrow w_1 \geq 1$

- From (2) and (3):
  $-(2w_1 + 2w_2 + b) \geq 1 \rightsquigarrow -2w_1 - 2w_2 - b \geq 1 \rightsquigarrow$
  $2w_2 \leq -2w_1 - b - 1$ && $2w_1 + b \geq 1 \Longrightarrow w_2 \leq -1$

Thus, $\frac{1}{2}\mathbf{w}^T\mathbf{w} = \frac{1}{2}(w_1^2 + w_2^2) \geq 1$ and the minimum is at $\mathbf{w} = (1, -1)$;
$(b = -1, w_1 = 1, w_2 = -1)$ satisfies the 4 constraints

The separating hyperplane H with maximum margin is given by $x_1 - x_2 - 1 = 0$.

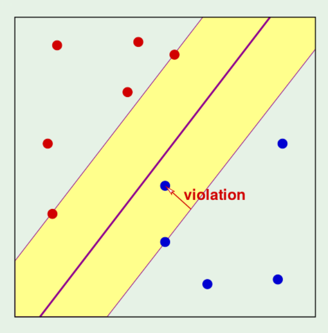
$$X = \begin{bmatrix} 0 & 0 \\ 2 & 2 \\ 2 & 0 \\ 3 & 0 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix}$$



The margin is $\frac{1}{||\mathbf{w}||} = \frac{1}{\sqrt{2}} \approx 0.707$

Non-linearly separable case

This case is dealt by considering a **soft margin** formulation as opposed to the (previuous) **hard margin** formulation:



Soft margin: $y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1 - \xi_n$

(**Hard margin**: $y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1$)

Optimization problem

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{n=1}^{N}\xi_n$$

$$\text{subject to} \quad y_n\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b\right) \geq 1 - \xi_n \text{ for } n = 1, 2, \ldots, N;$$

$$\xi_n \geq 0 \text{ for } n = 1, 2, \ldots, N.$$

$C >= 0$ is an user-specified parameter; the larger it is, the smaller the allowed margin violation

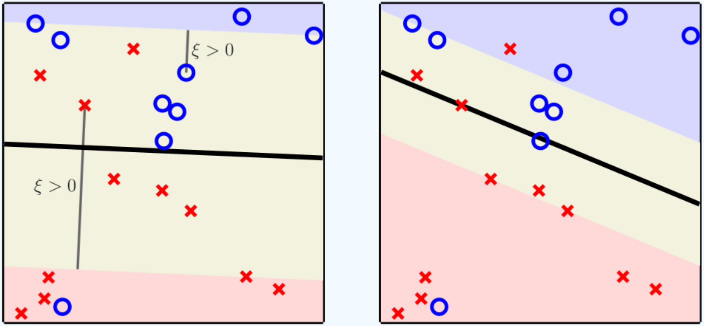
---

Compare to the hard-margin formulation:

$$\text{minimize} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

$$\text{subject to} \quad y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1, n = 1, \ldots, N$$

Intuition on constant C



(a) $C = 1$ (b) $C = 500$

$$\text{minimize } \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i$$

How to solve QP optimization problems ?

Both cases, hard and soft margin SVM, can be formulated as a QP optimization problem

Primal formulation: Standard QP optimization

Dual formulation: based on Lagrange formulation, dual QP

Standard QP optimization

Standard form of QP problems

M inequality constraints and Q positive semi-definite

$$
\begin{aligned}
\underset{\mathbf{u}}{\text{minimize}} \quad & \frac{1}{2}\mathbf{u}^T Q\mathbf{u} + \mathbf{p}^T\mathbf{u} \\
\text{subject to:} \quad & \mathbf{a}_m^T\mathbf{u} \geq c_m \quad (m = 1, \ldots, M)
\end{aligned}
$$

In matrix form

$$
\begin{aligned}
\underset{\mathbf{u}}{\text{minimize}} \quad & \frac{1}{2}\mathbf{u}^T Q\mathbf{u} + \mathbf{p}^T\mathbf{u} \\
\text{subject to:} \quad & A\mathbf{u} \geq \mathbf{c}
\end{aligned}
$$

QP solvers can be used to compute the optimal solution $\mathbf{u}^*$:

$$\mathbf{u}^* \leftarrow \mathrm{QP}(Q, \mathbf{p}, A, \mathbf{c})$$

SVM – standard QP formulation

QP problem formulation

$$\begin{aligned} \underset{\mathbf{u}}{\text{minimize}} \quad & \frac{1}{2}\mathbf{u}^T Q\mathbf{u} + \mathbf{p}^T\mathbf{u} \\ \text{subject to:} \quad & \mathbf{a}_m^T\mathbf{u} \geq c_m \\ & i = m, \ldots, M \end{aligned}$$

QP of hard-margin SVM

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} \\ \text{subject to:} \quad & y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1 \\ & i = 1, \ldots, N \end{aligned}$$

Denoting $\mathbf{u} = \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix}$, we have

$$\mathbf{w}^{\mathrm{T}}\mathbf{w} = \begin{bmatrix} b & \mathbf{w}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} 0 & \mathbf{0}_d^{\mathrm{T}} \\ \mathbf{0}_d & \mathrm{I}_d \end{bmatrix} \begin{bmatrix} b \\ \mathbf{w}^{\mathrm{T}} \end{bmatrix} = \mathbf{u}^{\mathrm{T}} \begin{bmatrix} 0 & \mathbf{0}_d^{\mathrm{T}} \\ \mathbf{0}_d & \mathrm{I}_d \end{bmatrix} \mathbf{u},$$

$$\mathbf{a}_n^{\mathrm{T}} = y_n \begin{bmatrix} 1 & \mathbf{x}_n^{\mathrm{T}} \end{bmatrix} \text{ and } c_n = 1$$

SVM – standard QP formulation

Linear Hard-Margin SVM with QP

1: Let $\mathbf{p} = \mathbf{0}_{d+1}$ ($(d+1)$-dimensional zero vector) and $\mathbf{c} = \mathbf{1}_N$ (N-dimensional vector of ones). Construct matrices Q and A, where

$$\mathrm{Q} = \begin{bmatrix} 0 & \mathbf{0}_d^{\mathrm{T}} \\ \mathbf{0}_d & \mathrm{I}_d \end{bmatrix}, \qquad \mathrm{A} = \underbrace{\begin{bmatrix} y_1 & -y_1\mathbf{x}_1^{\mathrm{T}}- \\ \vdots & \vdots \\ y_N & -y_N\mathbf{x}_N^{\mathrm{T}}- \end{bmatrix}}_{\text{signed data matrix}}.$$

2: Calculate $\begin{bmatrix} b^* \\ \mathbf{w}^* \end{bmatrix} = \mathbf{u}^* \leftarrow \mathsf{QP}(\mathrm{Q}, \mathbf{p}, \mathrm{A}, \mathbf{c})$.

3: Return the hypothesis $g(\mathbf{x}) = \mathrm{sign}(\mathbf{w}^{*\mathrm{T}}\mathbf{x} + b^*)$.

Dual QP optimization

Recall primal and dual formulation

When we discussed regularization, we started with the following optimization problem

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & E_{in}(\mathbf{w}) \\ \text{subject to:} \quad & \mathbf{w}^T\mathbf{w} \leq C \end{aligned}$$

and we ended up solving the following problem

$$\underset{\mathbf{w}}{\text{minimize}} \quad E_{aug}(\mathbf{w}) = E_{in}(\mathbf{w}) + \lambda \mathbf{w}^T\mathbf{w}$$

That is, we started with a problem with constraints on $\mathbf{w}$ and ended up with a problem without such contraints
(only one contraint: $\lambda \geq 0$)

Introduction to the dual formulation, first with one constraint

QP with one contraint:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\mathbf{u}^T Q\mathbf{u} + \mathbf{p}^T\mathbf{u} \\ \text{subject to:} \quad & \mathbf{a}^T\mathbf{u} \geq c \quad \text{(one contraint)} \end{aligned}$$

Fact: If there is an optimal solution $\mathbf{u}^*$ for the above problem, then $\mathbf{u}^*$ is also an optimal solution of the following problem:

$$\text{minimize} \quad \frac{1}{2}\mathbf{u}^T Q\mathbf{u} + \mathbf{p}^T\mathbf{u} + \max_{\alpha \geq 0} \alpha(c - \mathbf{a}^T\mathbf{u})$$

Why? Since $\mathbf{a}^T\mathbf{u} \geq c \iff c - \mathbf{a}^T\mathbf{u} \leq 0$, then $\max_{\alpha \geq 0} \alpha(c - \mathbf{a}^T\mathbf{u}) = 0$

Dual formulation

$$\underset{\mathbf{u}}{\text{minimize}} \quad \frac{1}{2}\mathbf{u}^T Q\mathbf{u} + \mathbf{p}^T\mathbf{u} + \max_{\alpha\geq 0} \alpha(c - \mathbf{a}^T\mathbf{u})$$

The term $\alpha(c - \mathbf{a}^T\mathbf{u})$ forces $c - \mathbf{a}^T\mathbf{u}$ to stay negative – i.e., to satisfy the constraint $\mathbf{a}^T\mathbf{u} \geq c$ (because this helps to minimize the cost function). On the other hand, α is chosen so as to maximize $\alpha(c - \mathbf{a}^T\mathbf{u})$ (to avoid $c - \mathbf{a}^T\mathbf{u}$ going to $-\infty$)

There is no constraints
We have a min-max optimization problem

Dual formulation

Theorem 8.7 (KKT)**.** For a feasible convex QP-problem in *primal* form,

$$\begin{aligned} \underset{\mathbf{u}\in\mathbb{R}^L}{\text{minimize:}} \quad & \tfrac{1}{2}\mathbf{u}^{\mathrm{T}}\mathrm{Q}\mathbf{u} + \mathbf{p}^{\mathrm{T}}\mathbf{u} \\ \text{subject to:} \quad & \mathbf{a}_m^{\mathrm{T}}\mathbf{u} \ge c_m \qquad (m = 1, \cdots, M), \end{aligned}$$

define the Lagrange function

$$\mathcal{L}(\mathbf{u}, \boldsymbol{\alpha}) = \tfrac{1}{2}\mathbf{u}^{\mathrm{T}}\mathrm{Q}\mathbf{u} + \mathbf{p}^{\mathrm{T}}\mathbf{u} + \sum_{m=1}^{M} \alpha_m \left(c_m - \mathbf{a}_m^{\mathrm{T}}\mathbf{u}\right).$$

The solution $\mathbf{u}^*$ is optimal for the primal if and only if $(\mathbf{u}^*, \boldsymbol{\alpha}^*)$ is a solution to the dual optimization problem

$$\max_{\boldsymbol{\alpha}\ge\mathbf{0}} \min_{\mathbf{u}} \mathcal{L}(\mathbf{u}, \boldsymbol{\alpha}).$$

KKT conditions

The optimal $(\mathbf{u}^*, \boldsymbol{\alpha}^*)$ satisfies the Karush-Kühn-Tucker (KKT) conditions:

(i) *Primal and dual constraints:*

$$\mathbf{a}_m^{\mathrm{T}} \mathbf{u}^* \geq c_m \qquad \text{and} \qquad \alpha_m \geq 0 \qquad\qquad (m = 1, \cdots, M).$$

(ii) *Complementary slackness:*

$$\alpha_m^* \left(\mathbf{a}_m^{\mathrm{T}} \mathbf{u}^* - c_m \right) = 0.$$

(iii) *Stationarity with respect to* $\mathbf{u}$*:*

$$\nabla_{\mathbf{u}} \mathcal{L}(\mathbf{u}, \boldsymbol{\alpha}) \big|_{\mathbf{u}=\mathbf{u}^*, \boldsymbol{\alpha}=\boldsymbol{\alpha}^*} = \mathbf{0}.$$

Solving the dual QP optimization problems

Dual: characterized by the Lagrangean $\mathcal{L}$

It is a min-max problem

$$\min_{\mathbf{u}} \max_{\alpha \geq 0} \mathcal{L}(\mathbf{u}, \alpha) = \max_{\alpha \geq 0} \min_{\mathbf{u}} \mathcal{L}(\mathbf{u}, \alpha)$$

Iterative optimization:

1. We fix α and optimize $\min_{\mathbf{u}} \mathcal{L}(\mathbf{u}, \alpha)$
2. Then, we fix $\mathbf{u}$ and optimize $\max_{\alpha} \mathcal{L}(\mathbf{u}, \alpha)$

Dual formulation for the hard-margin SVM

Primal formulation:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} \\ \text{subject to} \quad & y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1, n = 1, \ldots, N \end{aligned}$$

Lagrangean function

$$\begin{aligned} \mathcal{L}(b, \mathbf{w}, \boldsymbol{\alpha}) &= \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \sum_{n=1}^{N} \alpha_n \left(1 - y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)\right) \\ &= \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} - \sum_{n=1}^{N} \alpha_n y_n \mathbf{w}^{\mathrm{T}}\mathbf{x}_n - b\sum_{n=1}^{N} \alpha_n y_n + \sum_{n=1}^{N} \alpha_n. \end{aligned}$$

Solution of the dual hard-margin SVM

$$\begin{aligned}\mathcal{L}(b, \mathbf{w}, \boldsymbol{\alpha}) &= \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \sum_{n=1}^{N} \alpha_n \left(1 - y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)\right) \\ &= \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} - \sum_{n=1}^{N} \alpha_n y_n \mathbf{w}^{\mathrm{T}}\mathbf{x}_n - b\sum_{n=1}^{N} \alpha_n y_n + \sum_{n=1}^{N} \alpha_n.\end{aligned}$$

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

$$\frac{\partial \mathcal{L}}{\partial b} = -\sum_{n=1}^{N} \alpha_n y_n \quad \text{and} \quad \frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{w} - \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n.$$

Computing the zero:

$$\sum_{n=1}^{N} \alpha_n y_n = 0; \qquad \mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n.$$

Solution of the dual hard-margin SVM

$$\begin{aligned}\mathcal{L}(b, \mathbf{w}, \boldsymbol{\alpha}) &= \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \sum_{n=1}^{N} \alpha_n \left(1 - y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)\right) \\ &= \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} - \sum_{n=1}^{N} \alpha_n y_n \mathbf{w}^{\mathrm{T}}\mathbf{x}_n - b\sum_{n=1}^{N} \alpha_n y_n + \sum_{n=1}^{N} \alpha_n.\end{aligned}$$

$$\sum_{n=1}^{N} \alpha_n y_n = 0; \qquad \mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n.$$

The Lagrangean is reduced to a function on α only:

$$\mathcal{L}(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} y_n y_m \alpha_n \alpha_m \mathbf{x}_n^{\mathrm{T}}\mathbf{x}_m + \sum_{n=1}^{N} \alpha_n.$$

Solution of the dual hard-margin SVM

The Lagrangean is reduced to a function on α only:

$$\mathcal{L}(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} y_n y_m \alpha_n \alpha_m \mathbf{x}_n^{\mathrm{T}} \mathbf{x}_m + \sum_{n=1}^{N} \alpha_n .$$

2. Now we would like to maximize $\mathcal{L}$: Minimize $-\mathcal{L}$ with respect to α

$$\begin{aligned} \underset{\boldsymbol{\alpha}\in\mathbb{R}^N}{\text{minimize:}} \quad & \frac{1}{2}\sum_{m=1}^{N}\sum_{n=1}^{N} y_n y_m \alpha_n \alpha_m \mathbf{x}_n^{\mathrm{T}} \mathbf{x}_m - \sum_{n=1}^{N} \alpha_n \\ \text{subject to:} \quad & \sum_{n=1}^{N} y_n \alpha_n = 0 \\ & \alpha_n \geq 0 \qquad (n = 1, \cdots, N). \end{aligned}$$

This is a standard QP that can be solved using Solvers!

Solution of the dual hard-margin SVM

Minimization of $-\mathcal{L}$ with respect to α is a standard QP:

$$\begin{aligned} \underset{\boldsymbol{\alpha}\in\mathbb{R}^N}{\text{minimize:}} \quad & \tfrac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\mathrm{Q}_{\mathrm{D}}\boldsymbol{\alpha} - \mathbf{1}_N^{\mathrm{T}}\boldsymbol{\alpha} \\ \text{subject to:} \quad & \mathrm{A}_{\mathrm{D}}\boldsymbol{\alpha} \geq \mathbf{0}_{N+2}, \end{aligned} \qquad (8.$$

where Q_{D} and A_{D} (D for dual) are given by:

$$\mathrm{Q}_{\mathrm{D}} = \begin{bmatrix} y_1y_1\mathbf{x}_1^{\mathrm{T}}\mathbf{x}_1 & \dots & y_1y_N\mathbf{x}_1^{\mathrm{T}}\mathbf{x}_N \\ y_2y_1\mathbf{x}_2^{\mathrm{T}}\mathbf{x}_1 & \dots & y_2y_N\mathbf{x}_2^{\mathrm{T}}\mathbf{x}_N \\ \vdots & \vdots & \vdots \\ y_Ny_1\mathbf{x}_N^{\mathrm{T}}\mathbf{x}_1 & \dots & y_Ny_N\mathbf{x}_N^{\mathrm{T}}\mathbf{x}_N \end{bmatrix} \text{ and } \mathrm{A}_{\mathrm{D}} = \begin{bmatrix} \mathbf{y}^{\mathrm{T}} \\ -\mathbf{y}^{\mathrm{T}} \\ \mathrm{I}_{N\times N} \end{bmatrix}$$

$$\alpha^* \leftarrow QP(Q_D, -\mathbf{1}, A_D, \mathbf{0})$$

Hard-margin SVM with dual QP

Hard-Margin SVM with Dual QP

1: Construct Q_D and A_D as in Exercise 8.11

$$Q_D = \begin{bmatrix} y_1 y_1 \mathbf{x}_1^T \mathbf{x}_1 & \dots & y_1 y_N \mathbf{x}_1^T \mathbf{x}_N \\ y_2 y_1 \mathbf{x}_2^T \mathbf{x}_1 & \dots & y_2 y_N \mathbf{x}_2^T \mathbf{x}_N \\ \vdots & \vdots & \vdots \\ y_N y_1 \mathbf{x}_N^T \mathbf{x}_1 & \dots & y_N y_N \mathbf{x}_N^T \mathbf{x}_N \end{bmatrix} \text{ and } A_D = \begin{bmatrix} \mathbf{y}^T \\ -\mathbf{y}^T \\ I_{N \times N} \end{bmatrix}.$$

2: Use a QP-solver to optimize the dual problem:

$$\boldsymbol{\alpha}^* \leftarrow \mathsf{QP}(Q_D, -\mathbf{1}_N, A_D, \mathbf{0}_{N+2}).$$

3: Let s be a support vector for which $\alpha_s^* > 0$. Compute b^*,

$$b^* = y_s - \sum_{\alpha_n^* > 0} y_n \alpha_n^* \mathbf{x}_n^T \mathbf{x}_s.$$

4: Return the final hypothesis

$$g(\mathbf{x}) = \text{sign}\left(\sum_{\alpha_n^* > 0} y_n \alpha_n^* \mathbf{x}_n^T \mathbf{x} + b^* \right).$$

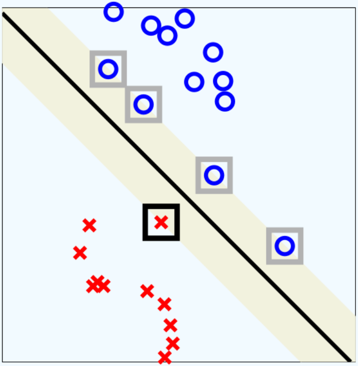
Interpretation of the solution

Support vectors: $\alpha_s > 0 \Longrightarrow y_s(\mathbf{w}^{*T}\mathbf{x}_s + b^*) = 1$

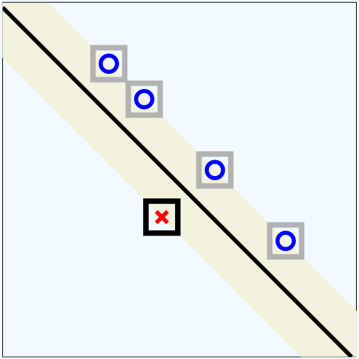
Weights: $\mathbf{w}^* = \sum_{n=1}^{N} y_n \alpha_n^* \mathbf{x}_n$

Bias: $b^* = y_s - \mathbf{w}^{*T}\mathbf{x}_s = y_s - \sum_{n=1}^{N} y_n \alpha_n^* \mathbf{x}_n$

Hipothesis: $g(\mathbf{x}) = \mathrm{sign}\left(\sum_{\alpha_n^* > 0} y_n \alpha_n^* \mathbf{x}_n^T \mathbf{x} + b^* \right)$



(a) All data



(b) Only support vectors

Solvers

— Our previous toy example —

— QP formulation with solution: `svm_cvxpy.ipynb` —

using CVXPY https://www.cvxpy.org/

Solution of the soft-margin SVM

The soft-margin SVM is also a QP

Only with more contraints

Thus the same discussion on QP and dual QP holds

The new optimization

Minimize $\quad \frac{1}{2}\,\mathbf{w}^{\intercal}\mathbf{w} + C\sum_{n=1}^{N}\xi_n$

subject to $\quad y_n\left(\mathbf{w}^{\intercal}\mathbf{x}_n + b\right) \geq 1 - \xi_n \quad$ for $\quad n = 1, \ldots, N$

and $\quad \xi_n \geq 0 \quad$ for $\quad n = 1, \ldots, N$

$$\mathbf{w} \in \mathbb{R}^d \;\;,\;\; b \in \mathbb{R} \;\;,\;\; \boldsymbol{\xi} \in \mathbb{R}^N$$

Lagrange formulation

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\,\mathbf{w}^{\intercal}\mathbf{w} + C\sum_{n=1}^{N}\xi_n - \sum_{n=1}^{N}\alpha_n(y_n\left(\mathbf{w}^{\intercal}\mathbf{x}_n + b\right) - 1 + \xi_n) - \sum_{n=1}^{N}\beta_n\,\xi_n$$

Minimize w.r.t. $\mathbf{w}$, b, and $\boldsymbol{\xi}$ and maximize w.r.t. each $\alpha_n \geq 0$ and $\beta_n \geq 0$

$$\nabla_{\mathbf{w}}\mathcal{L} = \mathbf{w} - \sum_{n=1}^{N}\alpha_n y_n \mathbf{x}_n = \mathbf{0}$$

$$\frac{\partial\mathcal{L}}{\partial b} = -\sum_{n=1}^{N}\alpha_n y_n = 0$$

$$\frac{\partial\mathcal{L}}{\partial \xi_n} = C - \alpha_n - \beta_n = 0$$

Optimization of $\mathcal{L}$ with respect to α:

Hard-margin

$$\begin{aligned} \underset{\boldsymbol{\alpha} \in \mathbb{R}^N}{\text{minimize:}} \quad & \frac{1}{2} \sum_{m=1}^{N} \sum_{n=1}^{N} y_n y_m \alpha_n \alpha_m \mathbf{x}_n^{\mathrm{T}} \mathbf{x}_m - \sum_{n=1}^{N} \alpha_n \\ \text{subject to:} \quad & \sum_{n=1}^{N} y_n \alpha_n = 0 \\ & \alpha_n \geq 0 \qquad\qquad (n = 1, \cdots, N). \end{aligned}$$

Soft-margin (também é um problema QP)

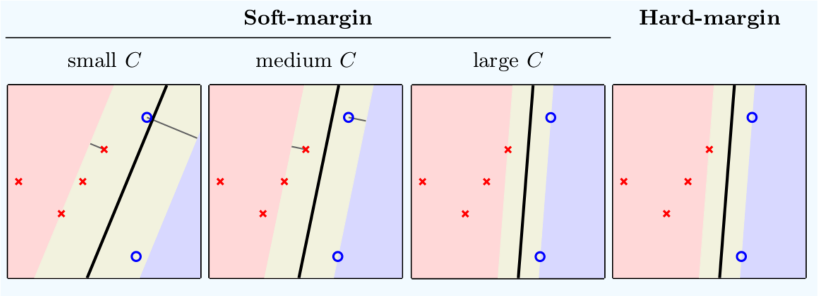
$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2} \boldsymbol{\alpha}^{\mathrm{T}} \mathrm{Q_D} \boldsymbol{\alpha} - \mathbf{1}^{\mathrm{T}} \boldsymbol{\alpha} \\ \text{subject to} \quad & \mathbf{y}^{\mathrm{T}} \boldsymbol{\alpha} = 0; . \\ & \mathbf{0} \leq \boldsymbol{\alpha} \leq C \cdot \mathbf{1}. \end{aligned}$$

Interpretation of C

$$\text{minimize } \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i$$

$$O \leq \alpha_n \leq C$$

Interpretation of C



$0 < \alpha_n^* < C \implies \mathbf{x}_n$ is a support vector
$\alpha_n^* = 0 \implies \mathbf{x}_n$ is beyond the margin on the right side
$\alpha_n^* = C \implies \mathbf{x}_n$ is in the margin or in the wrong side

When the data is linearly separable, there exists C such that the soft-margin SVM solution is exactly the same solution of the hard-margin SVM

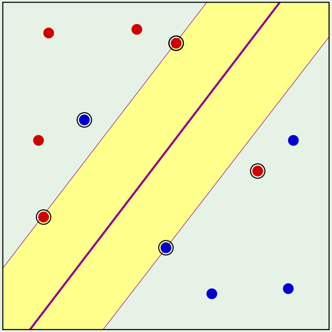
Types of support vectors

margin support vectors $\quad (0 < \alpha_n < C)$

$$y_n \left(\mathbf{w}^\intercal \mathbf{x}_n + b\right) = 1 \qquad (\xi_n \; = \; 0)$$

non-margin support vectors $\quad (\alpha_n = C)$

$$y_n \left(\mathbf{w}^\intercal \mathbf{x}_n + b\right) < 1 \qquad (\xi_n \; > \; 0)$$

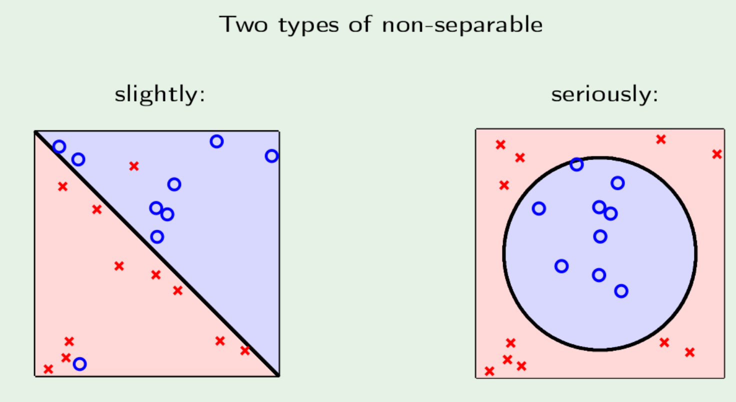


The Kernel trick

Motivation

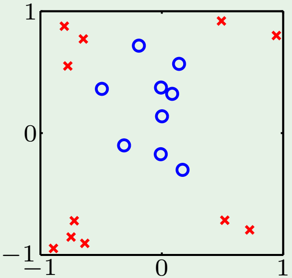
Soft-margin SVM could be used to solve non-linear cases

Would we get good solutions for both examples below ?

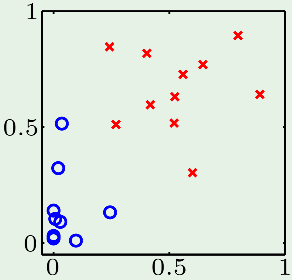


z instead of x

$$\mathcal{L}(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m \, \alpha_n \alpha_m \, \mathbf{z}_n^{\mathsf{T}} \mathbf{z}_m$$



$\mathcal{X} \longrightarrow \mathcal{Z}$



What is the problem?

When we map data $\mathbf{x} \in \mathbb{R}^d$ to $\mathbf{z} \in \mathbb{R}^{d'}$, $\tilde{d} >> d$, we may face computational problems

What do we need from the $\mathcal{Z}$ space?

$$\mathcal{L}(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m \, \alpha_n \alpha_m \, \mathbf{z}_n^{\intercal} \mathbf{z}_m$$

Constraints: $\alpha_n \geq 0$ for $n = 1, \cdots, N$ and $\sum_{n=1}^{N} \alpha_n y_n = 0$

$$\boxed{g(\mathbf{x}) = \operatorname{sign}\left(\mathbf{w}^{\intercal}\mathbf{z} + b\right)}$$ need $\mathbf{z}_n^{\intercal}\mathbf{z}$

where $\mathbf{w} = \sum_{\mathbf{z}_n \text{ is SV}} \alpha_n y_n \mathbf{z}_n$

and b: $y_m\left(\mathbf{w}^{\intercal}\mathbf{z}_m + b\right) = 1$ need $\mathbf{z}_n^{\intercal}\mathbf{z}_m$

Kernel trick

Is there any kernel function $K()$ satisfying

$$K_\Phi(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x})^T \Phi(\mathbf{x}')$$

and such that computation is more efficient than computing $\mathbf{z}^T\mathbf{z}' = \Phi(\mathbf{x})^T\Phi(\mathbf{x}')$?

If there is suck $K()$, then

$$\mathrm{Q_D} = \begin{bmatrix} y_1y_1\mathbf{x}_1^{\mathrm{T}}\mathbf{x}_1 & \dots & y_1y_N\mathbf{x}_1^{\mathrm{T}}\mathbf{x}_N \\ y_2y_1\mathbf{x}_2^{\mathrm{T}}\mathbf{x}_1 & \dots & y_2y_N\mathbf{x}_2^{\mathrm{T}}\mathbf{x}_N \\ \vdots & \vdots & \vdots \\ y_Ny_1\mathbf{x}_N^{\mathrm{T}}\mathbf{x}_1 & \dots & y_Ny_N\mathbf{x}_N^{\mathrm{T}}\mathbf{x}_N \end{bmatrix} \quad \mathrm{Q_D} = \begin{bmatrix} y_1y_1\mathrm{K}_{11} & \dots & y_1y_N\mathrm{K}_{1N} \\ y_2y_1\mathrm{K}_{21} & \dots & y_2y_N\mathrm{K}_{2N} \\ \vdots & \vdots & \vdots \\ y_Ny_1\mathrm{K}_{N1} & \dots & y_Ny_N\mathrm{K}_{NN} \end{bmatrix}$$

Kernel K would be equivalent to mapping $\mathbf{x}$ to $\mathbf{z}$ and applying dual SVM on $\mathbf{z}$, but without explicitly computing $\mathbf{z}$!

Hard-Margin SVM with Kernel

1: Construct $\mathrm{Q_D}$ from the kernel K, and $\mathrm{A_D}$:

$$\mathrm{Q_D} = \begin{bmatrix} y_1y_1\mathrm{K}_{11} & \dots & y_1y_N\mathrm{K}_{1N} \\ y_2y_1\mathrm{K}_{21} & \dots & y_2y_N\mathrm{K}_{2N} \\ \vdots & \vdots & \vdots \\ y_Ny_1\mathrm{K}_{N1} & \dots & y_Ny_N\mathrm{K}_{NN} \end{bmatrix} \quad \text{and} \quad \mathrm{A_D} = \begin{bmatrix} \mathbf{y}^{\mathrm{T}} \\ -\mathbf{y}^{\mathrm{T}} \\ \mathrm{I}_{N\times N} \end{bmatrix},$$

where $\mathrm{K}_{mn} = K(\mathbf{x}_m, \mathbf{x}_n)$. (K is called the *Gram* matrix.)

2: Use a QP-solver to optimize the dual problem:

$$\boldsymbol{\alpha}^* \leftarrow \mathsf{QP}(\mathrm{Q_D}, -\mathbf{1}_N, \mathrm{A_D}, \mathbf{0}_{N+2}).$$

3: Let s be any support vector for which $\alpha_s^* > 0$. Compute

$$b^* = y_s - \sum_{\alpha_n^*>0} y_n\alpha_n^* K(\mathbf{x}_n, \mathbf{x}_s).$$

4: Return the final hypothesis

$$g(\mathbf{x}) = \operatorname{sign}\left(\sum_{\alpha_n^*>0} y_n\alpha_n^* K(\mathbf{x}_n, \mathbf{x}) + b^* \right).$$

The final hypothesis

Express $g(\mathbf{x}) = \mathrm{sign}\left(\mathbf{w}^\intercal \mathbf{z} + b\right)$ in terms of $K(-,-)$

$$\mathbf{w} = \sum_{\mathbf{z}_n \text{ is SV}} \alpha_n y_n \mathbf{z}_n \implies g(\mathbf{x}) = \mathrm{sign}\left(\sum_{\alpha_n > 0} \alpha_n y_n K(\mathbf{x}_n, \mathbf{x}) + b\right)$$

where $b = y_m - \sum_{\alpha_n > 0} \alpha_n y_n K(\mathbf{x}_n, \mathbf{x}_m)$

for any support vector $(\alpha_m > 0)$

Examples of kernel

- Linear: $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}'$

- Polynomial of order Q: $K(\mathbf{x}, \mathbf{x}') = (\zeta + \gamma\mathbf{x}^T\mathbf{x}')^Q$, $\zeta, \gamma > 0$

- Gaussian RBF: $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma||\mathbf{x} - \mathbf{x}'||^2)$, $\gamma > 0$

Polynomial kernel

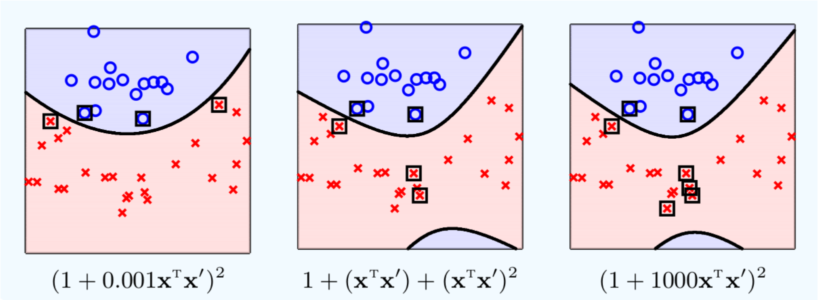
$\mathbf{x} = (x_1, \ldots, x_d)$
$\mathbf{z} = \Phi(\mathbf{x}) = (1, x_1, \ldots, x_d, x_1\, x_1, x_1\, x_2, \ldots, x_2\, x_1, \ldots, \ldots, x_d\, x_d)$

Dimension of $\mathbf{z}$: $d' = 1 + d + d^2$

$$\begin{aligned}
\Phi(\mathbf{x})^T \Phi(\mathbf{x}') &= 1 + \sum_{i=1}^{d} x_i x_i' + \sum_{i=1}^{d} \sum_{j=1}^{d} x_i x_j\, x_i' x_j' \\
&= 1 + \mathbf{x}^T \mathbf{x}' + (\sum_{i=1}^{d} x_i x_i')(\sum_{j=1}^{d} x_j x_j') \\
&= 1 + \mathbf{x}^T \mathbf{x}' + (\mathbf{x}^T \mathbf{x}')^2 = (1 + \mathbf{x}^T \mathbf{x})^2
\end{aligned}$$

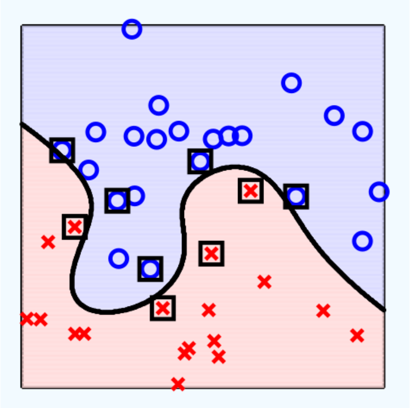
Computational complexity: from $\mathcal{O}(\tilde{d})$ to $\mathcal{O}(d)$

Example: polynomial kernel (degree 2)



$$K(\mathbf{x}, \mathbf{x}') = (\zeta + \gamma \mathbf{x}^T \mathbf{x}')^Q$$

Example: polynomial kernel (degree 10)



Gaussian-RBF kernel

$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma ||\mathbf{x} - \mathbf{x}'||^2)$ $(\gamma > 0)$

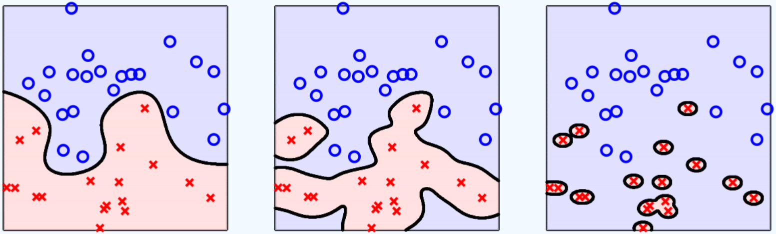
Expanding it for the case when $d = 1$

$$\begin{aligned} K(x, x') &= \exp\left(-\|x - x'\|^2\right) \\ &= \exp\left(-(x)^2\right) \cdot \exp(2xx') \cdot \exp\left(-(x')^2\right) \\ &= \exp\left(-(x)^2\right) \cdot \left(\sum_{k=0}^{\infty} \frac{2^k (x)^k (x')^k}{k!}\right) \cdot \exp\left(-(x')^2\right), \end{aligned}$$

$$\Phi(x) = \exp(-x^2) \cdot \left(1, \sqrt{\frac{2^1}{1!}}x, \sqrt{\frac{2^2}{2!}}x^2, \sqrt{\frac{2^3}{3!}}x^3, \ldots\right)$$

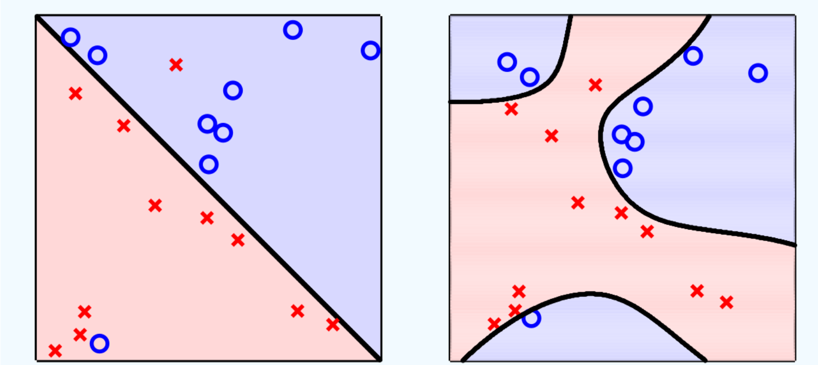
That means $d' = \infty$!

Example: Gaussian-RBF kernel



$\exp(-1\|\mathbf{x}-\mathbf{x}'\|^2)$ $\exp(-10\|\mathbf{x}-\mathbf{x}'\|^2)$ $\exp(-100\|\mathbf{x}-\mathbf{x}'\|^2)$

Example: linear $\times$ Gaussian-RBF kernels



(a) linear classifier (b) Gaussian-RBF kernel

How do we know that $\mathcal{Z}$ exists ...

... for a given $K(\mathbf{x}, \mathbf{x}')$? valid kernel

Three approaches:

1. By construction
2. Math properties (*Mercer's condition*)
3. Who cares? ☺

Design your own kernel

$K(\mathbf{x}, \mathbf{x}')$ is a valid kernel iff

1. It is symmetric and **2.** The matrix:

$$\begin{bmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & K(\mathbf{x}_1, \mathbf{x}_2) & \ldots & K(\mathbf{x}_1, \mathbf{x}_N) \\ K(\mathbf{x}_2, \mathbf{x}_1) & K(\mathbf{x}_2, \mathbf{x}_2) & \ldots & K(\mathbf{x}_2, \mathbf{x}_N) \\ \ldots & \ldots & \ldots & \ldots \\ K(\mathbf{x}_N, \mathbf{x}_1) & K(\mathbf{x}_N, \mathbf{x}_2) & \ldots & K(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}$$

is **positive semi-definite**

for any $\mathbf{x}_1, \cdots, \mathbf{x}_N$ (Mercer's condition)