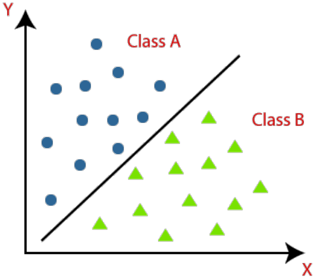
Support Vector Machine (SVM)

Introduction

Support Vector Machine (SVM):

- is a classification model.
- belongs to supervised learning.
- finds the global optimum, not a local optimum.

SVM is one of the **most effective** solutions for classification problem.

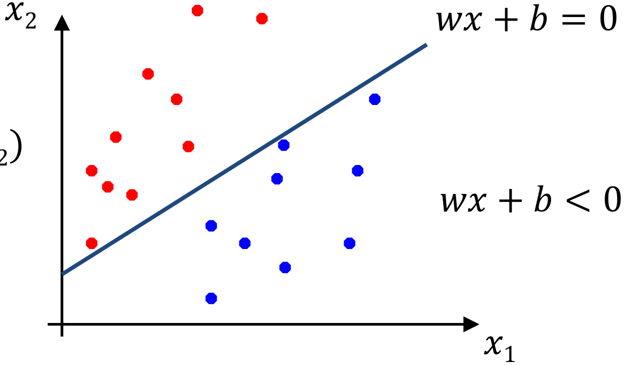


Introduction

Binary classification can be viewed as the task of separating classes in feature space.

$$wx + b > 0$$

$$w = (w_1, w_2)$$
$$x = (x_1, x_2)$$

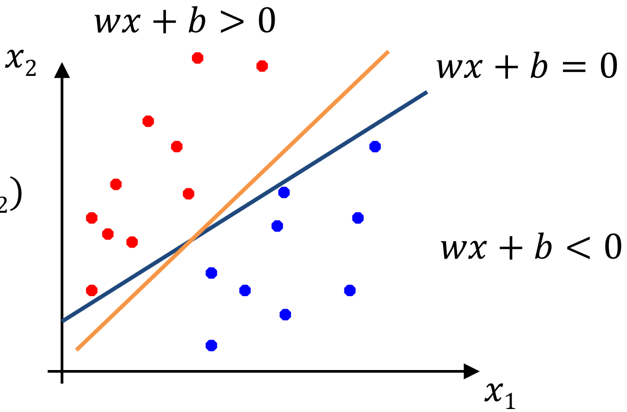


$$f(x) = sign(wx + b)$$

Introduction

There are infinite number of linear separators, which one is **optimal**?

$w = (w_1, w_2)$
$x = (x_1, x_2)$

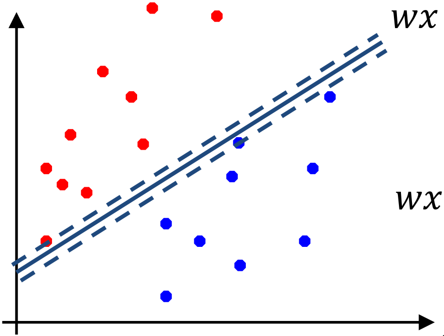


$$f(x) = sign(wx + b)$$

Introduction

There are infinite number of linear separators, which one is **optimal**?

$$wx + b > 0$$



$$wx + b = 0$$

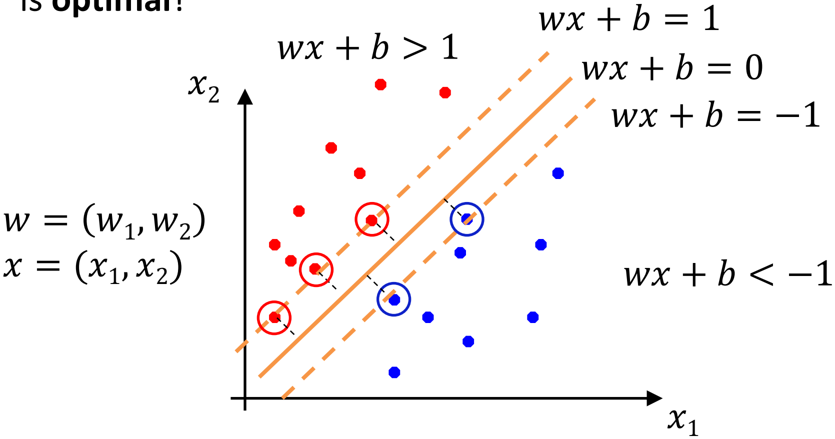
$$w = (w_1, w_2)$$
$$x = (x_1, x_2)$$

$$wx + b < 0$$

$$f(x) = sign(wx + b)$$

Introduction

There are infinite number of linear separators, which one is **optimal**?

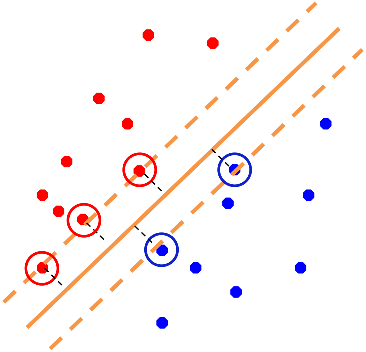


$$f(x) = sign(wx + b)$$

Training samples **on the margin lines** are called **support vectors** to shape the optimal hyperplane.

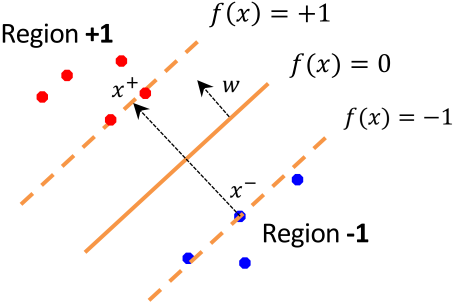
Introduction

There are infinite number of linear separators, which one is **optimal**?



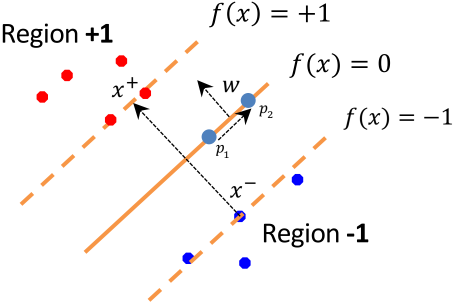
Training samples **on the margin lines** are called **support vectors** to shape the optimal hyperplane.

Introduction



Observation: Vector *w* is perpendicular to the decision boundary. Why?

Introduction



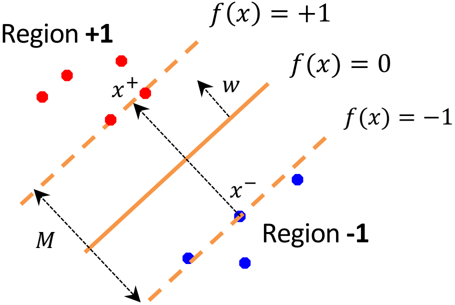
Vector *w* is perpendicular to the decision boundary, why?

Pick p_1 and p_2 on the decision boundary s.t. $f(p_1) = 0$ and $f(p_2) = 0$

We have $\begin{cases} wp_1 + b = 0 \\ wp_2 + b = 0 \end{cases}$, hence $w(p_1 - p_2) = 0$

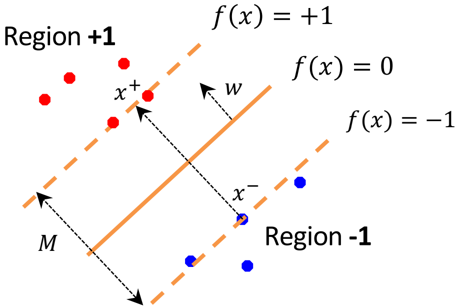
Therefore, $\vec{w} \perp \overrightarrow{p_1 p_2}$

Introduction



What is the width of separation margin M?

Introduction

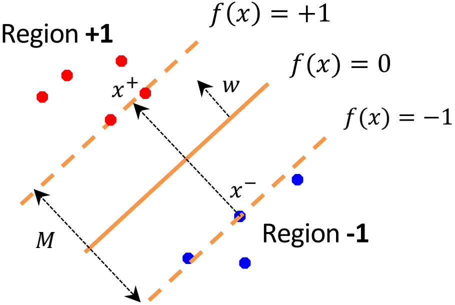


What is the width of separation margin M?

Pick x^- on a margin s.t. $f(x^-) = -1$; let x^+ be the closest point to x^- st $f(x^+) = 1$.

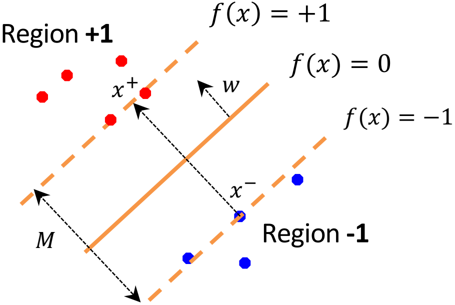
We have $\begin{cases} wx^+ + b = +1 \\ wx^- + b = -1 \end{cases}$, hence $w(x^+ - x^-) = +2 \rightarrow \mathrm{M} = \frac{2}{\|w\|}$

Introduction



Optimization: $\text{argmax}\ \mathrm{M} = \frac{2}{\|w\|} \rightarrow \text{argmin}\ \frac{\|w\|}{2} \rightarrow \text{argmin}\ \frac{1}{2}\|w\|^2$ such that all data points are on the correct side of the margins (positive and negative side).

Introduction



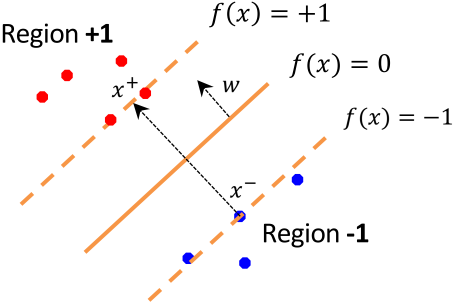
Problem:

$$\underset{w}{\operatorname{argmin}} \frac{1}{2} \|w\|^2$$

Constraint $\begin{cases} y^{(i)} = +1 \rightarrow wx^{(i)} + b \geq +1 \\ y^{(i)} = -1 \rightarrow wx^{(i)} + b \leq -1 \end{cases}$

This is a binary classification in which x is data feature and y is data label (+1 or -1).

Introduction



How to solve?

Problem:

$$\underset{w}{\operatorname{argmin}} \frac{1}{2}\|w\|^2$$

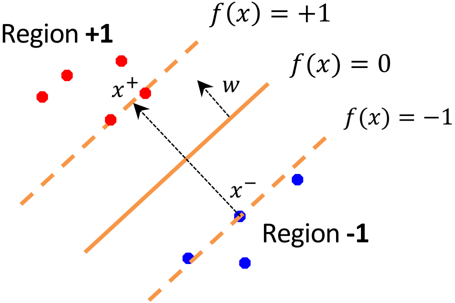
Constraint $\begin{cases} y^{(i)} = +1 \rightarrow wx^{(i)} + b \geq +1 \\ y^{(i)} = -1 \rightarrow wx^{(i)} + b \leq -1 \end{cases}$

Problem:

$$\underset{w}{\operatorname{argmin}} \frac{1}{2}\|w\|^2$$

Constraint $y^{(i)}(wx^{(i)} + b) \geq +1$

Introduction



Problem:

$$\underset{w}{\operatorname{argmin}} \frac{1}{2}\|w\|^2$$

Constraint $\begin{cases} y^{(i)} = +1 \rightarrow wx^{(i)} + b \geq +1 \\ y^{(i)} = -1 \rightarrow wx^{(i)} + b \leq -1 \end{cases}$

How to solve?

- Karush-Kuhn-Tucker (KKT) conditions.

Problem:

$$\underset{w}{\operatorname{argmin}} \frac{1}{2}\|w\|^2$$

Constraint $y^{(i)}(wx^{(i)} + b) \geq +1$

Exercise: Optimization problem

$$\textit{Find } \operatorname*{argmin}_{x,y} f(x,y) = x^2 + y^2$$
$$\textit{Subject to } g(x,y) = x + y = 1$$

<u>High school solution?</u>

Exercise: Optimization problem

$$\text{Find } \operatorname*{argmin}_{x,y} f(x,y) = x^2 + y^2$$
$$\text{Subject to } g(x,y) = x + y = 1$$

High school solution

$$f(x) = x^2 + (1-x)^2 = 2x^2 - 2x + 1$$

$$f'(x) = 4x - 2 = 0 \to x = \frac{1}{2} \to y = \frac{1}{2} \to f_{min}\left(\frac{1}{2}, \frac{1}{2}\right) = \frac{1}{2}$$

Exercise: Optimization problem

$$\textit{Find} \ \underset{x,y}{\operatorname{argmin}} f(x,y) = x^2 + y^2$$
$$\textit{Subject to} \ g(x,y) = x + y = 1$$

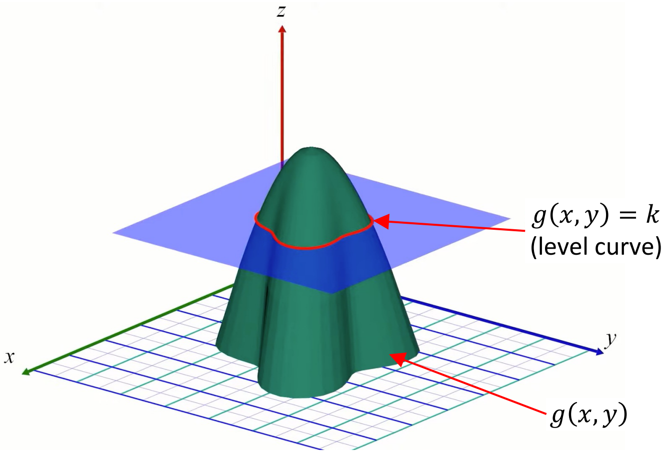
University solution

Use **Lagrange multiplier** technique.

Joseph-Louis Lagrange.
(Italian mathematician and astronomer)

Exercise: Optimization problem

$$Find\ \underset{x,y}{\operatorname{argmin}} f(x,y)$$
$$Subject\ to\ g(x,y) = k$$



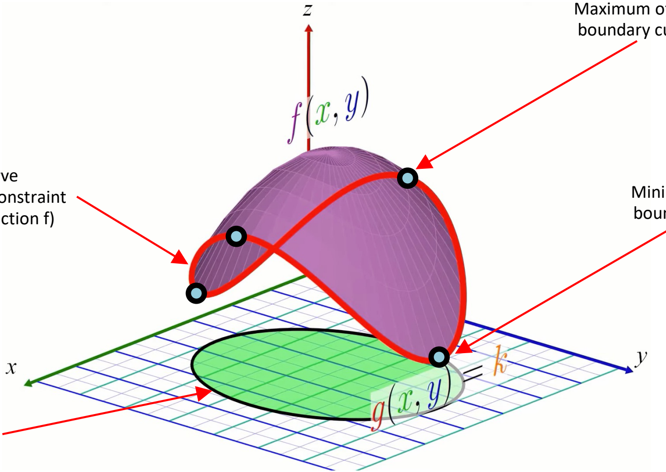
Exercise: Optimization problem

$$\textit{Find} \ \underset{x,y}{\operatorname{argmin}} f(x,y)$$
$$\textit{Subject to} \ g(x,y) = k$$

Maximum of the boundary curve

Boundary curve
(Projection of the constraint curve g on the function f)

Minimum of the boundary curve

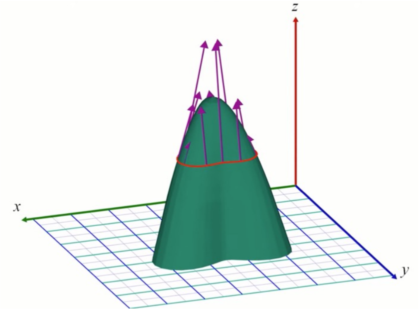


Constraint curve
(level curve)
$g(x,y) = k$

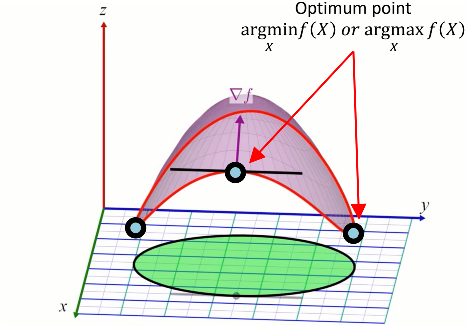
Lagrange multiplier (equality constraint)

$$Find \ \underset{X}{\operatorname{argmin}} f(X) \ or \ \underset{X}{\operatorname{argmax}} f(X)$$

$$Subject\ to \quad \begin{aligned} g_1(X) &= k_1 \\ g_2(X) &= k_2 \\ &\dots \\ g_n(X) &= k_n \end{aligned}$$



Gradient vectors of g are perpendicular to the **level curves** at points on the level curve.



Gradient vector of f is perpendicular to the **tangent line** of boundary curve at the max/min points.

Intuition: Gradient vectors lie along the direction of greatest change for a function, whereas:

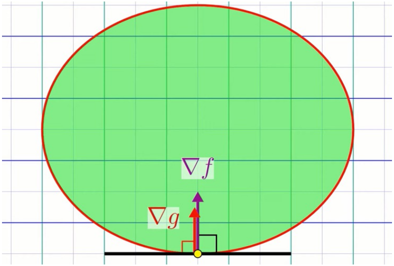
- **level curve** lies along the direction of *no change* (all points have the same value).
- **tangent line** is *flat* at the min/max points.

Since the gradient vectors should not "waste effort" to point in any direction of the level curve or the tangent line. Therefore they are perpendicular to the level curves and the tangent line

Lagrange multiplier (equality constraint)

$$Find\ \underset{X}{\operatorname{argmin}} f(X)\ or\ \underset{X}{\operatorname{argmax}} f(X)$$

$$Subject\ to\quad \begin{array}{l} g_1(X) = k_1 \\ g_2(X) = k_2 \\ \ldots \\ g_n(X) = k_n \end{array}$$



Gradient vector ∇f is parallel to ∇g as they are perpendicular to the tangent line that the point X.

Solution

$$\boxed{\begin{array}{l} \nabla f(X) = \sum_{i=1}^{n} \lambda_i \nabla g_i(X) \\ g_i(X) = k_i \end{array}}$$

which means setting $\nabla \mathcal{L}(x, y, \lambda) = 0$

$\mathcal{L}(x, y, \lambda) = f(X) - \sum_{i=1}^{n} \lambda_i\, g_i(X)$ is **Lagrange function**.

λ is **Lagrange multiplier**.

Exercise: Optimization problem

$$\mathit{Find}\ \underset{x,y}{\operatorname{argmin}} f(x,y) = x^2 + y^2$$
$$\mathit{Subject\ to}\ g(x,y) = x + y = 1$$

University solution

Use **Lagrange multiplier** technique?

Solution

$$\nabla f(X) = \sum_{i=1}^{n} \lambda_i \nabla g_i(X)$$

$$g_i(X) = k_i$$

Lagrange multiplier (equality constraint)

$$\mathit{Find}\ \underset{x,y}{\operatorname{argmin}} f(x,y) = x^2 + y^2$$
$$\mathit{Subject\ to}\ g(x,y) = x + y = 1$$

Solution using Lagrange multiplier

$$\begin{cases} \nabla f(x,y) = \lambda \nabla g(x,y) \\ x + y = 1 \end{cases}$$

$$\rightarrow \begin{cases} \left\langle \dfrac{\partial f}{\partial x}, \dfrac{\partial f}{\partial y} \right\rangle = \left\langle \lambda \dfrac{\partial g}{\partial x}, \lambda \dfrac{\partial g}{\partial y} \right\rangle \\ x + y = 1 \end{cases} \rightarrow \begin{cases} \langle 2x, 2y \rangle = \langle \lambda, \lambda \rangle \\ x + y = 1 \end{cases} \rightarrow \begin{cases} x = \lambda/2 \\ y = \lambda/2 \\ x + y = 1 \end{cases} \rightarrow \begin{cases} x = 1/2 \\ y = 1/2 \\ \lambda = 1 \end{cases}$$

$$\rightarrow f_{min}\left(\frac{1}{2}, \frac{1}{2}\right) = \frac{1}{2}$$

Karush-Kuhn-Tucker (KKT, inequality constraint)

KKT condition is a generalization of Lagrange Multiplier (equality constraints) to deal with inequality constraints. Given the problem:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & h_i(x) \leq 0, \quad i = 1, \dots m \\ & \ell_j(x) = 0, \quad j = 1, \dots r \end{aligned}$$

The ***Karush–Kuhn–Tucker conditions (KKT)*** are:

$$\nabla f(x) - \sum_{i=1}^{m} \lambda_i \nabla h_i(x) - \sum_{j=1}^{r} \mu_j \nabla l_j(x) = 0 \qquad \text{(stationary)}$$

$$h_i(x) \leq 0 \qquad \text{(primal feasibility)}$$

$$l_j(x) = 0 \qquad \text{(primal feasibility)}$$

$$\lambda_i \leq 0 \qquad \text{(dual feasibility)}$$

$$\lambda_i h_i(x) = 0 \qquad \text{(complementary slackness)}$$

Karush-Kuhn-Tucker (KKT, inequality constraint)

Problem: Given,

$$
\begin{aligned}
f(x,y) &= x^3 + y^2 \\
g(x,y) &= x^2 - 1 \geq 0
\end{aligned}
$$

$$
\boxed{\begin{aligned}
g(x) \geq 0 &\Rightarrow \lambda \geq 0 \\
g(x) \leq 0 &\Rightarrow \lambda \leq 0 \\
g(x) = 0 &\Rightarrow \lambda \text{ is unconstrained}
\end{aligned}}
$$

Find the extreme values.

Step 1: solve gradient of the Lagrangian

$$
\begin{aligned}
L(x,y,\lambda) &= f(x,y) - \lambda g(x,y) \\
&= x^3 + y^2 - \lambda(x^2 - 1)
\end{aligned}
$$

$$\nabla L(x,y,\lambda) = \nabla f(x,y) - \lambda \nabla g(x,y) = 0$$

$$
\left\{
\begin{aligned}
\frac{\partial}{\partial x} L(x,y,\lambda) &= 3x^2 - 2\lambda x = 0 \\
\frac{\partial}{\partial y} L(x,y,\lambda) &= 2y = 0 \\
\frac{\partial}{\partial \lambda} L(x,y,\lambda) &= x^2 - 1 = 0
\end{aligned}
\right.
$$

$x = \pm 1, \lambda = \pm \frac{3}{2}$

Step 2: check with constraint $\lambda \geq 0$

$\lambda = \frac{3}{2}, x = 1, y = 0, f(x,y) = 1$

which exactly means the condition $g(x,y) \geq 0$

Applying KKT to SVM – Primal form of SVM

Quadratic function to minimize, thus there is a single global minimum

Problem:

$$\underset{w}{\operatorname{argmin}} \frac{1}{2}\|w\|^2$$

Constraint $y^{(i)}(wx^{(i)} + b) \geq +1$

$\xrightarrow{\text{Rewrite}}$

Primal form of SVM:

$$\underset{w}{\operatorname{argmin}}\, f(w) = \frac{1}{2}\|w\|^2$$

s.t. $g_i(w, b) = y_i(wx_i + b) - 1 \geq 0$

Inequality constraint

Applying KKT to SVM – Primal form of SVM

Primal form of SVM:

$$\underset{w}{\operatorname{argmin}} f(w) = \frac{1}{2}\|w\|^2$$

$$\text{s.t. } g_i(w, b) = y_i(wx_i + b) - 1 \geq 0$$

Lagrangian function:

$$L(w, b, \lambda) = \frac{1}{2}\|w\|^2 - \sum_i \lambda_i \overbrace{(y_i(wx_i + b) - 1)}^{>= 0}$$

Hence, the goal of primal form is equivalent to minimizing w and b, but maximizing λ.

$$\min_{w,b} \max_{\lambda} L(w, b, \lambda)$$

Furthermore, the Lagrangian duality is

$$\min_{w,b} \max_{\lambda} L(w, b, \lambda) = \max_{\lambda} \min_{w^*,b^*} L(w, b, \lambda)$$

Applying KKT to SVM – Primal form of SVM

Primal form of SVM:

$$\underset{w}{\operatorname{argmin}} f(w) = \frac{1}{2}\|w\|^2$$
$$\text{s.t. } g_i(w,b) = y_i(wx_i + b) - 1 \geq 0$$

Lagrangian function:

$$L(w,b,\lambda) = \frac{1}{2}\|w\|^2 - \sum_i \lambda_i(y_i(wx_i + b) - 1)$$

The KKT conditions to solve the primal form (or $\min_{w,b} L(w,b,\lambda)$) are:

$$\frac{\partial}{\partial w} L(w,b,\lambda) = w - \sum_i \lambda_i y_i x_i = 0$$
$$\frac{\partial}{\partial b} L(w,b,\lambda) = -\sum_i \lambda_i y_i = 0$$
$$y_i(wx_i + b) - 1 \geq 0$$
$$\lambda_i \geq 0$$
$$\lambda_i(y_i(wx_i + b) - 1) = 0$$

Optimal solution:

$$w^* = \sum_i \lambda_i y_i x_i$$

and $\sum_i \lambda_i y_i = 0$

and $\lambda_i \geq 0$

Applying KKT to SVM – Primal form of SVM

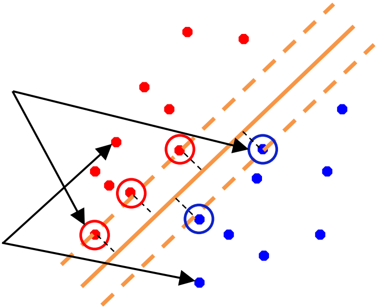
Let's analyze the last three conditions:

$$\left.\begin{aligned} y_i(wx_i + b) - 1 \geq 0 \\ \lambda_i \geq 0 \\ \lambda_i(y_i(wx_i + b) - 1) = 0 \end{aligned}\right\}$$

Observations:

- For points having $\boldsymbol{\lambda_i > 0}$, the $y_i(wx_i + b) - 1 = 0$ where $y_i = \pm 1$, which means $(wx_i + b) = \pm 1 = y_i$. These points **are on** the margins and are called support vectors.

- For points having $\boldsymbol{\lambda_i = 0}$, the constraint $y_i(wx_i + b) - 1 > 0$ where $y_i = \pm 1$, which means $(wx_i + b) > 1$ or $(wx_i + b) < 1$. These points **are not on** the margins.



Applying KKT to SVM – Primal form of SVM

Therefore, the optimal solution is:

$$\boxed{\begin{aligned} w^* &= \sum_i \lambda_i y_i x_i \\ b^* &= \frac{1}{y_k} - w^* x_k \end{aligned}}$$

$$with \sum_i \lambda_i y_i = 0$$
$$and\ \lambda_i \geq 0$$

where b^* can be calculated by using any data sample k where $\lambda_k > 0$, i.e., use a support vector k to compute b^*.

The weight w^* is a linear combination of the training inputs x_i, the training outputs y_i, and the values of λ_i. Most of the λ_i will turn out to have the value zero. The non-zero λ_i will correspond to the support vectors.

These are the first-order optimality conditions. Instead of directly calculating w^* and b^*, we substitute them back to the Lagrangian function to calculate λ s such that $\max_{\lambda} \min_{w^*, b^*} L(w, b, \lambda) = \min_{w,b} \max_{\lambda} L(w, b, \lambda)$, then calculate w^* and b^*.

A new example $\boldsymbol{x_{new}}$ has the prediction label $\boldsymbol{y_{new}} = \mathbf{sign}(\boldsymbol{w^* x_{new}} + \boldsymbol{b^*})$.

Dual Form of SVM

Primal form of SVM:

$$\underset{w}{\operatorname{argmin}} f(w) = \frac{1}{2}\|w\|^2$$
$$\text{s.t. } g_i(w, b) = y_i(w \cdot x_i + b) - 1 \geq 0$$

The Lagrangian Dual Problem: instead of minimizing over w, b subjecting to constraints involving λ, we can maximize over λ (the dual variable) subject to the relations obtained previously (optimality) for w and b.

The solution must satisfy the two relations:

$$w^* = \sum_i \lambda_i y_i x_i \qquad \sum_i \lambda_i y_i = 0$$

By substituting for w and b back in the original Lagrangian equation, we can get rid of the dependence on w and b.

Dual Form of SVM

Primal form of SVM:

$$\underset{w}{\operatorname{argmin}} f(w) = \frac{1}{2}\|w\|^2$$
$$\text{s.t. } g_i(w,b) = y_i(w \cdot x_i + b) - 1 \geq 0$$

Optimality:

$$w = \sum_i \lambda_i y_i x_i \quad \text{and} \quad \sum_i \lambda_i y_i = 0$$

$$\|w\|^2 = \sum_i \lambda_i y_i x_i \sum_j \lambda_j y_j x_j = \sum_i \sum_j \overbrace{\lambda_i \lambda_j y_i y_j}^{\text{Scalar}} \overbrace{\langle x_i^t, x_j \rangle}^{\text{Inner (dot) product}}$$

Lagrangian dual form of SVM:

$$\underbrace{\underset{\lambda}{\operatorname{argmax}}\, \underset{w,b}{\operatorname{argmin}}}_{\text{Dual problem}} L(w,b,\lambda) = \frac{1}{2}\|w\|^2 - \sum_i \lambda_i (y_i(w \cdot x_i + b) - 1)$$

$$s.t. \quad w = \sum_i \lambda_i y_i x_i \ \& \sum_i \lambda_i y_i = 0 \ \& \ \lambda_i \geq 0$$

Dual Form of SVM

Optimality: $$\|w\|^2 = \sum_i \lambda_i y_i x_i \sum_j \lambda_j y_j x_j = \sum_i \sum_j \lambda_i \lambda_j y_i y_j \langle x_i^t, x_j \rangle \qquad w = \sum_i \lambda_i y_i x_i \qquad \sum_i \lambda_i y_i = 0$$

$$\sum_i \lambda_i (y_i (w \cdot x_i + b) - 1) = \underbrace{\sum_i \lambda_i y_i \langle w, x_i \rangle} + b \overbrace{\sum_i \lambda_i y_i}^{\text{Zero}} - \sum_i \lambda_i$$

$$\sum_i \lambda_i y_i \left\langle \sum_j \lambda_j y_j x_j, x_i \right\rangle = \sum_i \sum_j \lambda_i \lambda_j y_i y_j \langle x_i^t, x_j \rangle$$

$$L(w, b, \lambda) = \frac{1}{2} \|w\|^2 - \sum_i \lambda_i (y_i (w \cdot x_i + b) - 1)$$

$$= \frac{1}{2} \sum_i \sum_j \lambda_i \lambda_j y_i y_j \langle x_i^t, x_j \rangle - \sum_i \sum_j \lambda_i \lambda_j y_i y_j \langle x_i^t, x_j \rangle - 0 + \sum_i \lambda_i$$

$$\boxed{L(\lambda) = \sum_i \lambda_i - \frac{1}{2} \sum_i \sum_j \lambda_i \lambda_j y_i y_j \langle x_i^t, x_j \rangle}$$ **(at optimality w, b)**

Dual Form of SVM

Optimality: $\|w\|^2 = \sum_i \lambda_i y_i x_i \sum_j \lambda_j y_j x_j = \sum_i \sum_j \lambda_i \lambda_j y_i y_j \langle x_i^t, x_j \rangle$ $\quad w = \sum_i \lambda_i y_i x_i \quad \sum_i \lambda_i y_i = 0$

$$\boxed{\begin{gathered} \underset{\lambda}{\operatorname{argmax}}\, L(\lambda) = \sum_i \lambda_i - \frac{1}{2} \sum_i \sum_j \lambda_i \lambda_j y_i y_j \langle x_i^t, x_j \rangle \\ s.t. \quad \sum_i \lambda_i y_i = 0 \ \& \ \lambda_i \geq 0 \end{gathered}}$$

Dual form gets rid of the dependence on $\boldsymbol{w}$ and $\boldsymbol{b}$

Quadratic programming techniques can be used to solve for $\boldsymbol{\lambda}$s, such as CVXOPT, ECOS, Gurobi, HiGHS, MOSEK, OSQP, ProxQP, qpOASES, qpSWIFT, quadprog, SCS, etc. Then $\boldsymbol{w}$ and $\boldsymbol{b}$ can be calculated.

Extra librabry practice: https://pypi.org/project/qpsolvers/

Quadratic programming

$$\underset{\lambda}{\operatorname{argmax}} L(\lambda) = \sum_i \lambda_i - \frac{1}{2}\sum_i \sum_j \lambda_i \lambda_j y_i y_j \langle x_i^t, x_j \rangle$$
$$s.t. \sum_i \lambda_i y_i = 0 \ \& \ \lambda_i \geq 0$$

Quadratic programming expects the optimization to be in the standard form of:

$$\underset{\lambda}{\operatorname{argmin}} L(\lambda) = \frac{1}{2}\lambda^T P \lambda + q^T \lambda$$
$$s.t. A\lambda = b \ \& \ G\lambda \leq h \ \& \ lb \leq \lambda \leq ub$$

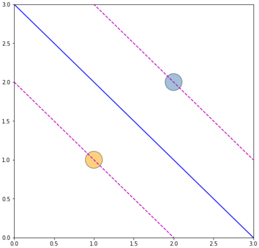
Need to convert the dual form optimization to the standard form:

$$\underset{\lambda}{\operatorname{argmin}} L(\lambda) = \frac{1}{2}\sum_i \sum_j \lambda_i \lambda_j y_i y_j \langle x_i^t, x_j \rangle - \sum_i \lambda_i = \underset{\lambda}{\operatorname{argmin}} = \frac{1}{2}\lambda^T P \lambda + q^T \lambda$$

For instance, assume that $X = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}$ and $y = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

$P = y_i y_j \langle x_i^t, x_j \rangle = (Y * Y^T).* (X * X^T) = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 \\ 4 & 8 \end{bmatrix} = \begin{bmatrix} 2 & -4 \\ -4 & 8 \end{bmatrix}$

$\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}; q = \begin{bmatrix} -1 \\ -1 \end{bmatrix}; G = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; h = 0; A = [1 \quad -1]; b = 0;$



Call the quadratic programming solver to find λ^*: $\text{solvers.qp}(P, q, G, h, A, b)$

$\lambda^* = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, w^* = \sum_i \lambda_i y_i x_i = 1 \times 1 \times \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 1 \times (-1) \begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$

$b^* = \frac{1}{y_k} - w^T x_k = \frac{1}{1} - [-1 \quad -1] \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 3$ or $= \frac{1}{-1} - [-1 \quad -1] \begin{bmatrix} 2 \\ 2 \end{bmatrix} = 3$. Hence $[-1 \quad -1] \begin{bmatrix} x^{(1)} \\ x^{(2)} \end{bmatrix} + 3 = 0$ is the equation of the boundary line.

Dual Form of SVM

$$\underset{\lambda}{\operatorname{argmax}} L(\lambda) = \sum_i \lambda_i - \frac{1}{2}\sum_i \sum_j \lambda_i \lambda_j y_i y_j \langle x_i^t, x_j \rangle$$

$$s.t. \quad \sum_i \lambda_i y_i = 0 \;\; \& \; \lambda_i \geq 0$$

- The solution λs we find here will be the same as the solution to the primal problem.
- This will let us solve the problem by computing the just the inner products $\langle x_i^t, x_j \rangle$ (which will be very important later on when we want to solve **non-linearly separable** classification problems) using Kernel Trick**.**

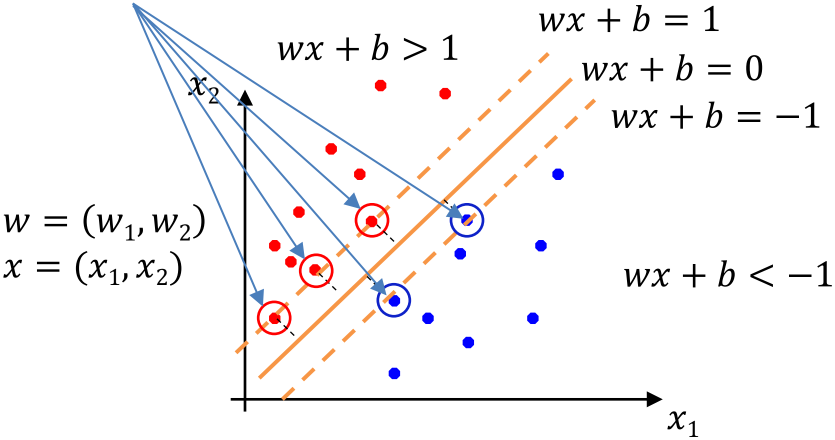
Dual Form of SVM

- **Primal:** $(w_1, w_2, \dots, w_d, b)$ → d+1 primal variables (d is the feature dimension)
- **Dual:** $\lambda_1, \lambda_2, \dots, \lambda_N$ → N dual variables (N is the number of data samples)

→ **Primal form** is preferred when we don't need to apply kernel trick to the data and the dataset is large but the dimension of each data point is small.

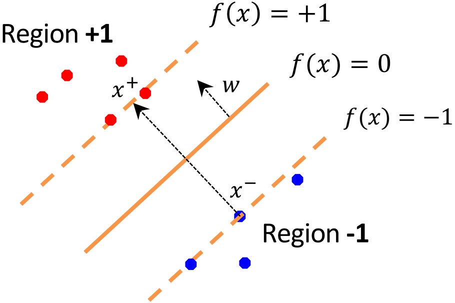
→ **Dual form** is preferred when data has a huge dimension and we need to apply the kernel trick.

Support vectors!!!



$$f(x) = sign(wx + b)$$

Hard Margin SVM



Primal form of SVM:

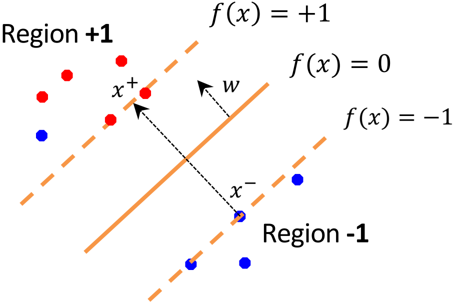
$$\underset{w}{\operatorname{argmin}}\, f(w) = \frac{1}{2}\|w\|^2$$
$$\text{s.t. } g_i(w, b) = y_i(w \cdot x_i + b) - 1 \geq 0$$

Hard margin SVM

Hard margin SVM can work only when data is **completely linearly separable** without any errors (noise or outliers).

→ Objective is to **maximize the margin**.

Soft Margin SVM



Primal form of SVM with soft margin:

$$\underset{w}{\operatorname{argmin}}\, f(w) = \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

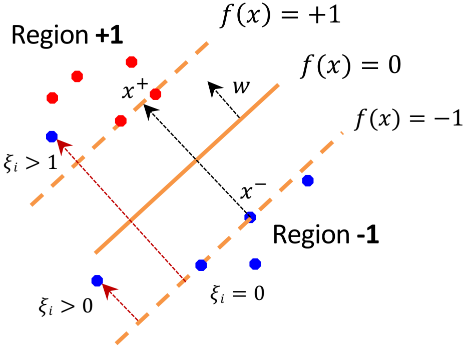
$$\text{s.t. } g_i(w,b) = y_i(w \cdot x_i + b) \geq 1 - \xi_i$$

$$\text{and } \xi_i \geq 0$$

Soft margin SVM

Soft margin SVM is when the dataset is noisy (with some overlap in positive and negative samples), there will be some error in classifying them with the hyper-plane
→ Objective is to **maximize the margin** along with **minimize the per-sample errors in classification**.

Soft Margin SVM



Primal form of SVM with soft margin:

$$\underset{w}{\operatorname{argmin}} f(w) = \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

s.t. $g_i(w,b) = y_i(w \cdot x_i + b) \geq 1 - \xi_i$

and $\xi_i \geq 0$

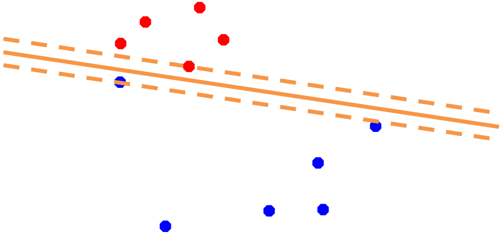
Soft margin

Soft margin SVM is when the dataset is noisy (with some overlap in positive and negative samples), there will be some error in classifying them with the hyper-plane

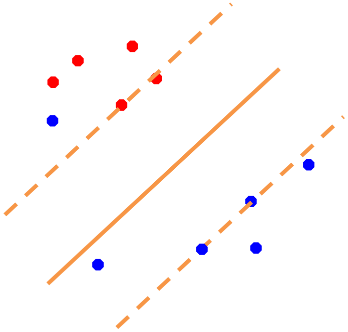
→ Objective is to **maximize the margin** along with **minimize the per-sample errors in classification** (like regularization to penalize misclassified points).

Hyper-parameter **C** tells how much we want to avoid misclassifying each training example.

Hard Margin vs. Soft Margin



Hard margin



Soft margin

- C tends to $+\infty$, what happens?
- C tends to 0, what happens?

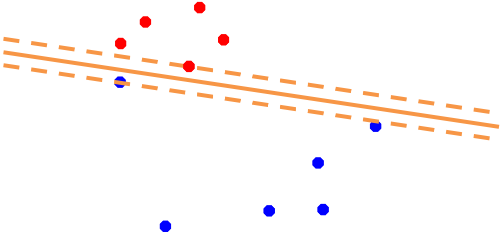
Primal form of SVM with soft margin:

$$\underset{w}{\operatorname{argmin}}\, f(w) = \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

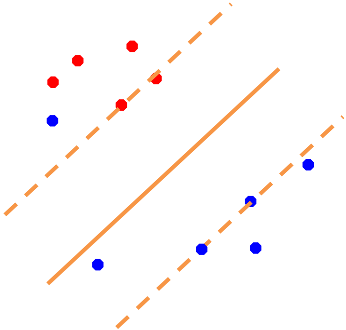
$$\text{s.t. } g_i(w,b) = y_i(w \cdot x_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

Hard Margin vs. Soft Margin



Hard margin



Soft margin

If C is large, we want very **few** misclassifications.

If C is **small**, we allow more misclassifications.

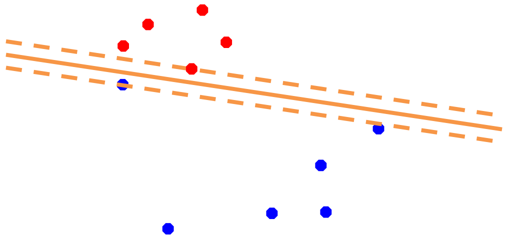
Primal form of SVM with soft margin:

$$\underset{w}{\operatorname{argmin}} f(w) = \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

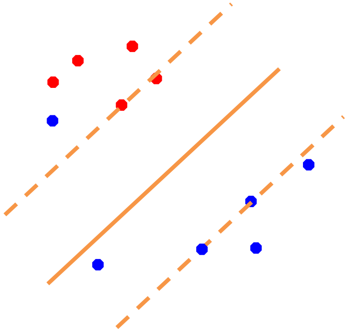
s.t. $g_i(w,b) = y_i(w \cdot x_i + b) \geq 1 - \xi_i$

and $\xi_i \geq 0$

Hard Margin vs. Soft Margin



Hard margin



Soft margin

Soft margin is a generalization of Hard margin!, but how to find C?

Primal form of SVM with soft margin:

$$\underset{w}{\operatorname{argmin}} f(w) = \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

$$\text{s.t. } g_i(w,b) = y_i(w \cdot x_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

Dual Form with Soft Margin

Primal form of SVM with soft margin:

$$\underset{w}{\operatorname{argmin}} f(w) = \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

$$\text{s.t. } g_i(w,b) = y_i(w \cdot x_i + b) \geq 1 - \xi_i$$

and $\xi_i \geq 0$

$$L(w,b,\lambda)$$

$$= \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

$$-\sum_i \lambda_i(y_i(w \cdot x_i + b) - 1 + \xi_i) - \sum_i \mu_i \xi_i$$

$$\frac{\partial}{\partial w} L(w,b,\xi) = w - \sum_i \lambda_i y_i x_i = 0$$

$$\frac{\partial}{\partial b} L(w,b,\xi) = -\sum_i \lambda_i y_i = 0$$

$$\frac{\partial}{\partial \xi_i} L(w,b,\xi) = C - \lambda_i - \mu_i = 0$$

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i$$

$$\lambda_i \geq 0, \mu_i \geq 0$$

$$\lambda_i(y_i(w \cdot x_i + b) - 1) = 0, \mu_i \xi_i = 0$$

Optimality:

$$w = \sum_i \lambda_i y_i x_i$$

and $\sum_i \lambda_i y_i = 0$

$$\mu_i = C - \lambda_i$$

$$\mu_i \geq 0 \rightarrow 0 \leq \lambda_i \leq C$$

Dual Form with Soft Margin

$$\mu_i = C - \lambda_i$$

Optimality: $\|w\|^2 = \sum_i \lambda_i y_i x_i \sum_j \lambda_j y_j x_j = \sum_i \sum_j \lambda_i \lambda_j y_i y_j \langle x_i, x_j \rangle$ $\quad w = \sum_i \lambda_i y_i x_i \quad \sum_i \lambda_i y_i = 0$

$$L(w,b,\lambda) = \frac{1}{2}\|w\|^2 + C\sum_i \xi_i - \sum_i \lambda_i (y_i(w \cdot x_i + b) - 1 + \xi_i) - \sum_i \mu_i \xi_i$$

$$= \frac{1}{2}\sum_i \sum_j \lambda_i \lambda_j y_i y_j \langle x_i, x_j \rangle + C\sum_i \xi_i - \left(\sum_i \lambda_i y_i w \cdot x_i + \overbrace{b\sum_i \lambda_i y_i}^{\text{Zero}} - \sum_i \lambda_i + \sum_i \lambda_i \xi_i\right) - \left(C\sum_i \xi_i - \sum_i \lambda_i \xi_i\right)$$

$$\boxed{L(\lambda) = \sum_i \lambda_i - \frac{1}{2}\sum_i \sum_j \lambda_i \lambda_j y_i y_j \langle x_i, x_j \rangle}$$ (w, b, μ_i, ξ_i are removed)

Dual Form with Soft Margin

$$\mu_i = C - \lambda_i$$

Optimality: $\|w\|^2 = \sum_i \lambda_i y_i x_i \sum_j \lambda_j y_j x_j = \sum_i \sum_j \lambda_i \lambda_j y_i y_j \langle x_i^t, x_j \rangle$ $\quad w = \sum_i \lambda_i y_i x_i \quad \sum_i \lambda_i y_i = 0$

$$L(w,b,\lambda) = \frac{1}{2}\|w\|^2 + C\sum_i \xi_i - \sum_i \lambda_i (y_i(w \cdot x_i + b) - 1 + \xi_i) - \sum_i \mu_i \xi_i$$

$$= \frac{1}{2}\sum_i \sum_j \lambda_i \lambda_j y_i y_j \langle x_i^t, x_j \rangle + C\sum_i \xi_i - \left(\underbrace{\sum_i \lambda_i y_i w \cdot x_i}_{\sum_i \sum_j \lambda_i \lambda_j y_i y_j \langle x_i^t, x_j \rangle} + b \overbrace{\sum_i \lambda_i y_i}^{\text{Zero}} - \sum_i \lambda_i + \sum_i \lambda_i \xi_i \right)$$
$$- \left(C\sum_i \xi_i - \sum_i \lambda_i \xi_i \right)$$

$$\boxed{L(\lambda) = \sum_i \lambda_i - \frac{1}{2}\sum_i \sum_j \lambda_i \lambda_j y_i y_j \langle x_i^t, x_j \rangle}$$

(w, b, μ_i, ξ_i are canceled out)

Dual Form with Soft Margin

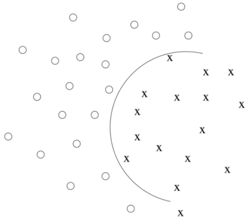
$$\underset{\lambda}{\operatorname{argmax}} L(\lambda) = \sum_i \lambda_i - \frac{1}{2}\sum_i \sum_j \lambda_i \lambda_j y_i y_j \langle x_i^t, x_j \rangle$$

$$s.t. \quad \sum_i \lambda_i y_i = 0 \;\; \& \; 0 \le \lambda_i \le C$$

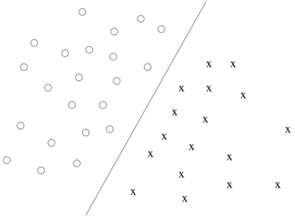
Dual form has **removed** the dependence on $\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\mu_i}, \boldsymbol{\xi_i}$

Use quadratic programming to solve for λs, then $\boldsymbol{w}$ and $\boldsymbol{b}$.

SVM Kernel

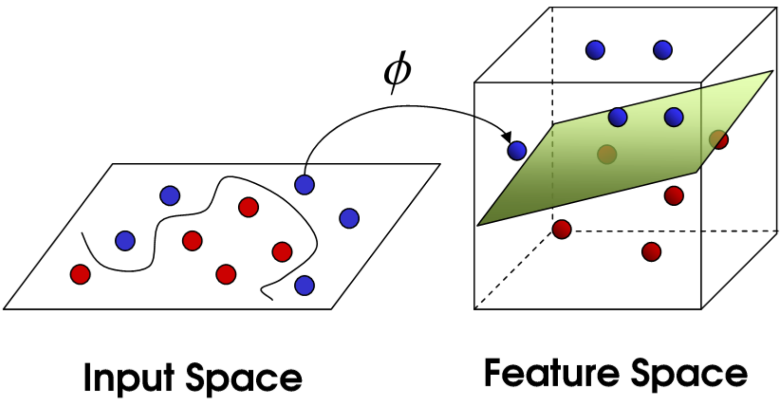


Non-linear separator in the original $\mathbf{x}$-space

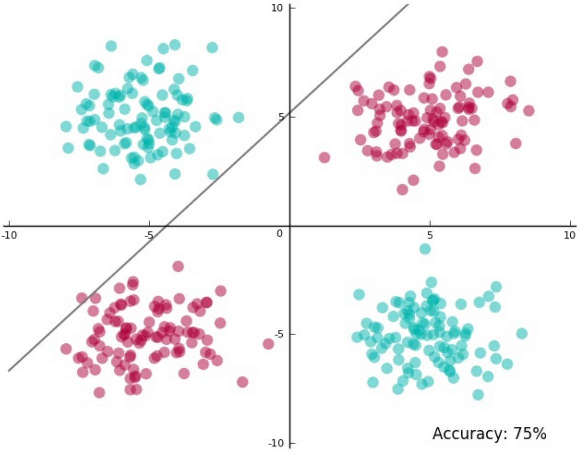


Linear separator in the feature ϕ-space

SVM Kernel

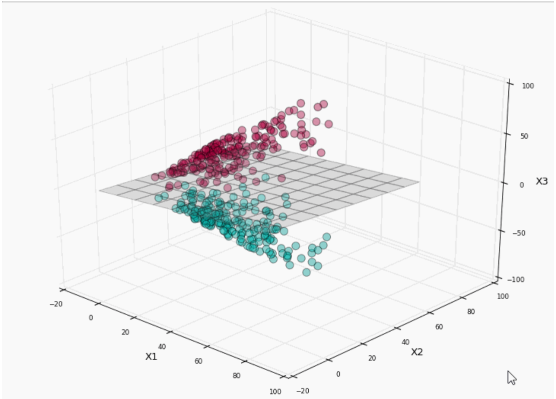


Polynomial Kernel



Polynomial Kernel

Project the 2-D feature (x_1, x_2) into 3-D feature $(x_1^2, x_2^2, \sqrt{2}x_1x_2)$.



3-D feature projection

SVM Kernel

A kernel function Φ takes as input two points in the **original space**, and **directly** gives us the **dot product** in the **projected space**.

Without Kernel trick

$$x = (x_1, x_2)$$
$$y = (y_1, y_2)$$
$$\Phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$$
$$\Phi(y) = (y_1^2, y_2^2, \sqrt{2}y_1y_2)$$
$$\Phi(x) \cdot \Phi(y) = \Phi_1(x)\Phi_1(y) + \Phi_2(x)\Phi_2(y) + \Phi_3(x)\Phi_3(y)$$

3 multiplications
+ 2 additions

4 multiplications

4 multiplications

13 elementary operations

SVM Kernel

A kernel kernel function Φ takes as input two points in the **original space**, and **directly** gives us the **dot product** in the **projected space**.

With Kernel trick

Define a Kernel function $K(x, y) = (x \cdot y)^2$

$$
\begin{aligned}
&K(x, y) \\
&= (x \cdot y)^2 \\
&= (x_1 y_1 + x_2 y_2)^2 \\
&= x_1^2 y_1^2 + x_2^2 y_2^2 + 2 x_1 x_2 y_1 y_2 \\
&= (x_1^2, x_2^2, \sqrt{2} x_1 x_2) \cdot (y_1^2, y_2^2, \sqrt{2} y_1 y_2) \\
&= \Phi(x) \cdot \Phi(y)
\end{aligned}
$$

4 elementary operations

Only 30% operation used!

SVM Kernel

Define a Kernel function $K(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{u} \cdot \boldsymbol{v})^d$

d=1

$$\phi(u).\phi(v) = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} . \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = u_1 v_1 + u_2 v_2 = u.v$$

d=2

$$\phi(u).\phi(v) = \begin{pmatrix} u_1^2 \\ u_1 u_2 \\ u_2 u_1 \\ u_2^2 \end{pmatrix} . \begin{pmatrix} v_1^2 \\ v_1 v_2 \\ v_2 v_1 \\ v_2^2 \end{pmatrix} \begin{aligned} &= u_1^2 v_1^2 + 2 u_1 v_1 u_2 v_2 + u_2^2 v_2^2 \\ &= (u_1 v_1 + u_2 v_2)^2 \\ &= (u.v)^2 \end{aligned}$$

For any *d (we will skip proof):*

$$\phi(u).\phi(v) = (u.v)^d$$

Taking a dot product and exponentiating gives same results as mapping into high dimensional space and then taking dot product.

SVM Kernel

A kernel kernel function Φ takes as input two points in the **original space**, and **directly** gives us the **dot product** in the **projected space**.

$$\underset{\lambda\in[0,C]}{\operatorname{argmax}} L(\lambda) = \sum_i \lambda_i - \frac{1}{2}\sum_i \sum_j \lambda_i \lambda_j y_i y_j \left\langle \Phi(x_i), \Phi\left(x_j\right)\right\rangle$$

$$\text{s.t.} \sum_i \lambda_i y_i = 0$$

It is computationally **expensive** to calculate $\langle\Phi(x),\Phi(u)\rangle$ in high-dimensional feature space.
→ Apply Kernel trick.

$$w = \sum_i \lambda_i y_i \Phi(x_i)$$

$$y(u) = \operatorname{sign}\left(\sum_i \lambda_i y_i \langle\Phi(x),\Phi(u)\rangle + b\right)$$

SVM Kernel

Define a Kernel function $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{y}) = \langle \Phi(x), \Phi(u) \rangle$

$$\underset{\lambda \in [0,C]}{\operatorname{argmax}} L(\lambda) = \sum_i \lambda_i - \frac{1}{2} \sum_i \sum_j \lambda_i \lambda_j y_i y_j \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{y})$$

$$\text{s.t.} \sum_i \lambda_i y_i = 0$$

$$w = \sum_i \lambda_i y_i \Phi(x_i)$$

$$y(u) = \operatorname{sign}\left(\sum_i \lambda_i y_i \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{u}) + b \right)$$

Apply Kernel trick.
→ **Less expensive**
Never compute features explicitly → Compute dot products in closed form

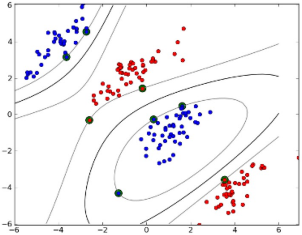
SVM Kernel

Define a Kernel function $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{y}) = \langle \Phi(x), \Phi(u) \rangle$

Linear: $K(x_i, x_j) = x_i \cdot x_j$

Polynomial of power p: $K\left(x_i, x_j\right) = (x_i \cdot x_j + 1)^p$

Gaussian: $K(x_i, xj) = exp^{\left(-\|x_i - x_j\|^2 / 2\sigma^2\right)}$



SVM Kernel Overfitting

- Huge feature space with kernels: should we worry about overfitting?
  - SVM objective seeks a solution with large **margin**
    - Theory says that large margin leads to good generalization (we will see this in a couple of lectures)
  - But everything overfits sometimes!!!
  - Can control by:
    - Setting C
    - Choosing a better Kernel
    - Varying parameters of the Kernel (width of Gaussian, etc.)

Pros and cons of SVM

Pros	Cons
• Work well with an optimal decision boundary. • Effective in high dimensional space. • Only use a subset of training samples (i.e., support vectors) to make prediction decision → Memory efficient.	• Slow training time when the dataset is large. • Does not directly provide the classification probability estimate.

Summary

- Support Vector machine
- KKT condition
- Primal form
- Dual form
- Soft margin vs. Hard margin
- Kernel trick

Q&A

Thank you