January 16, 2018

Inter-University Research Institute Corporation /
Research Organization of Information and Systems
National Institute of Informatics

Support Vector Machines

Data Mining 10 (データマイニング)

Mahito Sugiyama (杉山麿人)

Today's Outline

- Today's topic is support vector machines (SVMs)
  - A popular supervised classification method
- Perform binary classification by maximizing the margin
- Kernel trick for nonlinear classification

Classification Problem Setting

- Given a supervised dataset $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_N, y_N)\}$, $\boldsymbol{x}_i \in \mathbb{R}^n$ (feature vector), $y_i \in C = \{-1, 1\}$ (label)

- Use a decision function (hyperplane) in the form of

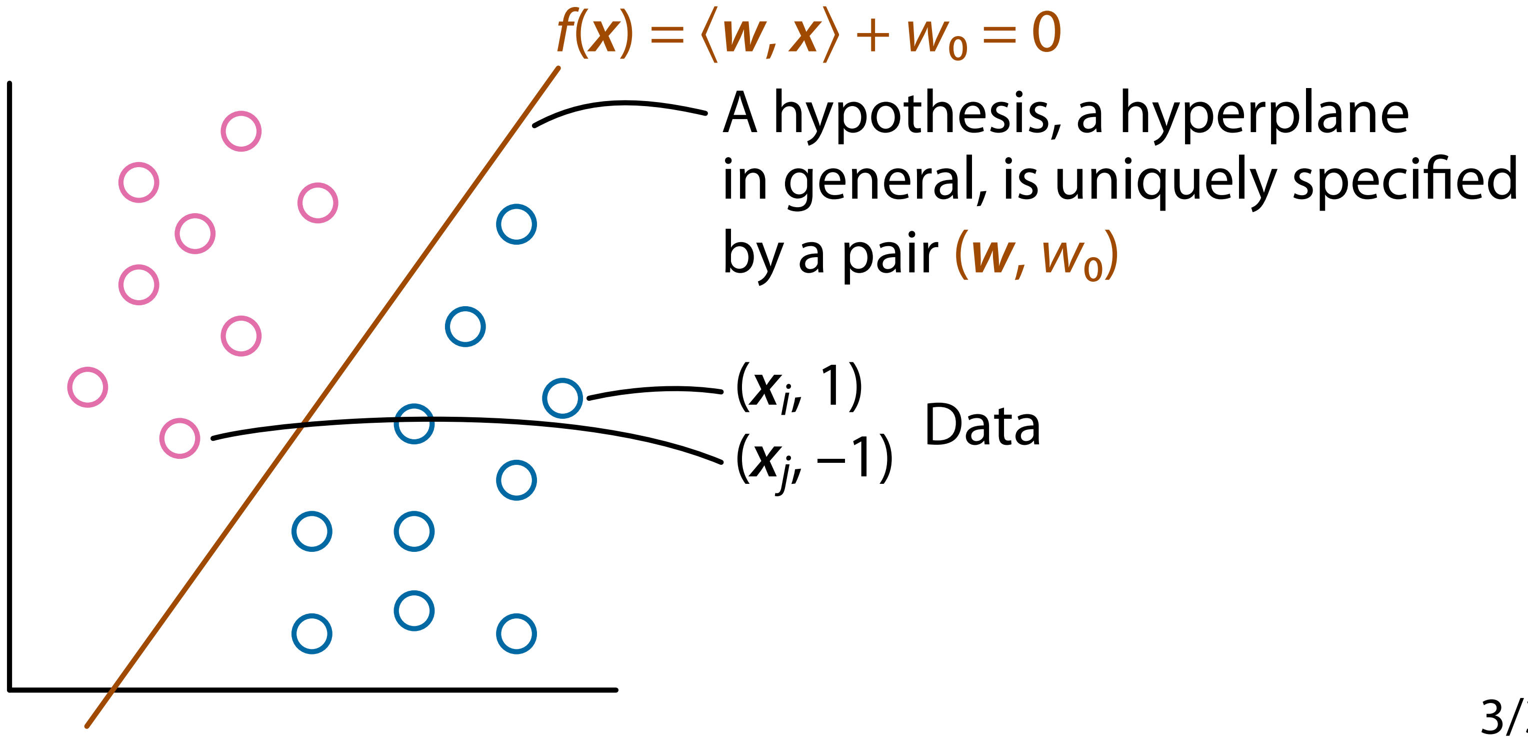
  $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + w_0 = \sum_{j=1}^{n} w^j x^j + w_0$

- A classifier $g(\boldsymbol{x})$ is given as

  $$g(\boldsymbol{x}) = \begin{cases} 1 & \text{if } f(\boldsymbol{x}) > 0, \\ -1 & \text{if } f(\boldsymbol{x}) < 0 \end{cases}$$

- Goal: Find $(\boldsymbol{w}, w_0)$ that correctly classifies the dataset

Classification by Hyperplane



Example: Perceptron

Algorithm 1: Perceptron

perceptron(D)

  Set small random values to $\boldsymbol{w}$ and w_o; // initialization

  foreach $i \in \{1, 2, \ldots, N\}$ **do**

    $a \leftarrow \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + w_o$

    if $y_i \cdot a < 0$ **then**

      $\boldsymbol{w} \leftarrow \boldsymbol{w} + y_i \boldsymbol{x}_i$; // update the weight

      $w_o \leftarrow w_o + y_i$; // update the bias

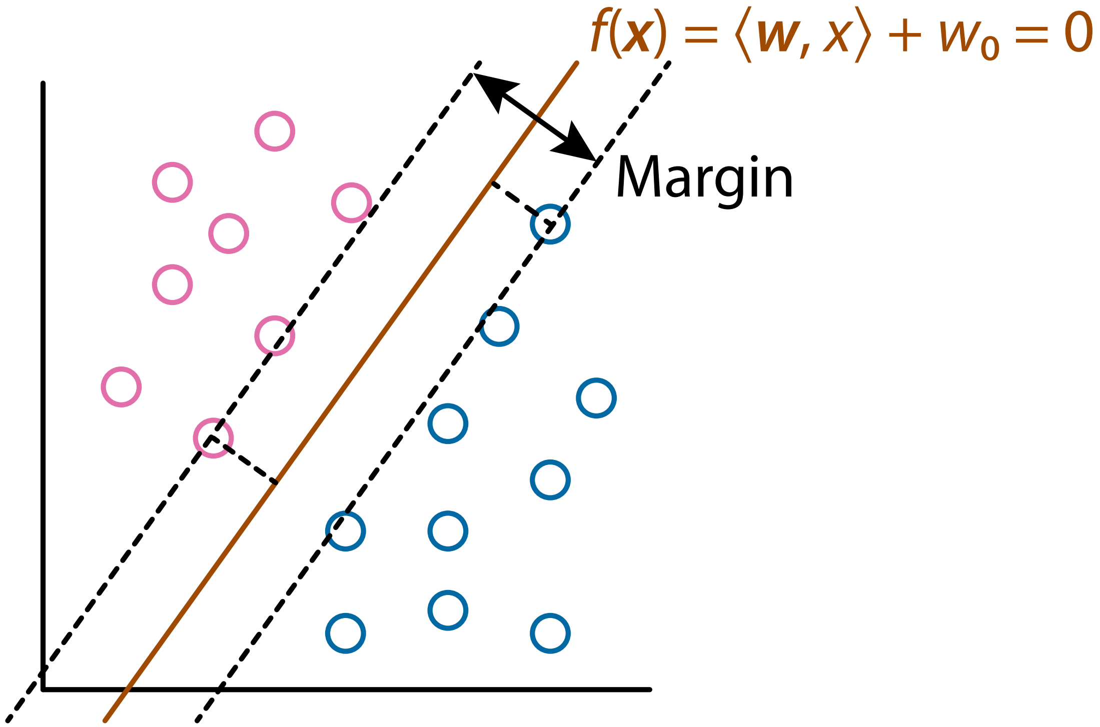
Correctness of Perceptron

- It is guaranteed that a perceptron always converges to a correct classifier
  - A correct classifier is a function f s.t.
    $f(\boldsymbol{x}) > 0$ if $y = 1$,
    $f(\boldsymbol{x}) < 0$ if $y = -1$
  - The convergence theorem
- Note: there are (infinitely) many functions that correctly classify two classes
  - A perceptron converges to one of them

Support Vector Machines (SVMs)

- A dataset D is separable by $f \iff y_i f(\boldsymbol{x}_i) > 0, \forall i \in \{1, 2, \ldots, N\}$
- The margin is the distance from the classification hyperplane to the closest data point
- Support vector machines (SVMs) tries to find a hyperplane that maximize the margin

Margin



Formulation of SVMs

- The distance from a point $\boldsymbol{x}_i$ to a hyperplane $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + w_0$ is
$$\frac{|f(\boldsymbol{x}_i)|}{||\boldsymbol{w}||} = \frac{|\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + w_0|}{||\boldsymbol{w}||}$$
- Since $y_i f(\boldsymbol{x}_i) > 0$ should be satisfied, assume that there exists $M > 0$ such that $y_i f(\boldsymbol{x}_i) \geq M$ for all $i \in \{1, 2, \dots, N\}$
- The margin maximization problem can be written as
$$\max_{\boldsymbol{w}, w_0, M} \frac{M}{||\boldsymbol{w}||} \quad \text{subject to } y_i f(\boldsymbol{x}_i) \geq M, i \in \{1, 2, \dots, N\}$$
  - $M = \min_{i \in \{1,2,\dots,N\}} |\langle \boldsymbol{w}, x_i \rangle + w_0|$

Hard Margin SVMs

- We can eliminate M and obtain

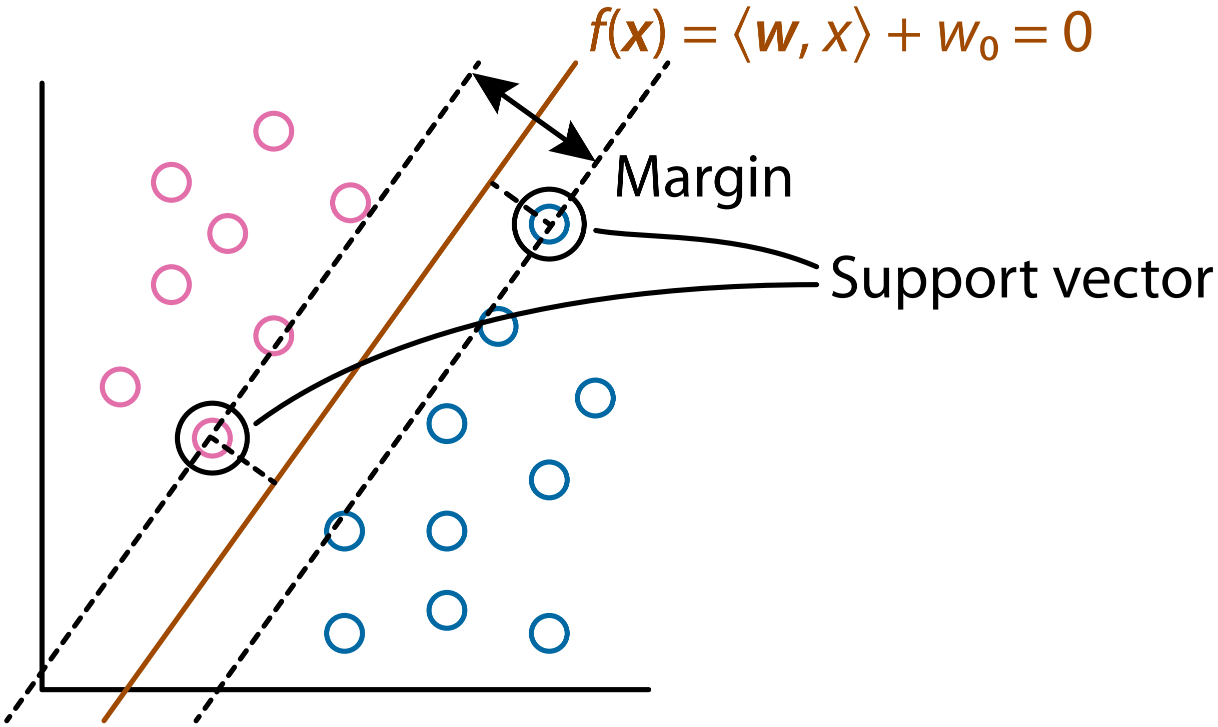
$$\max_{\boldsymbol{w}, w_0} \frac{1}{||\boldsymbol{w}||} \quad \text{subject to } y_i f(\boldsymbol{x}_i) \geq 1, i \in \{1, 2, \ldots, N\}$$

- This is equivalent to

$$\min_{\boldsymbol{w}, w_0} ||\boldsymbol{w}||^2 \quad \text{subject to } y_i f(\boldsymbol{x}_i) \geq 1, i \in \{1, 2, \ldots, N\}$$

  - The standard formulation of hard margin SVMs
  - There are data points x_i satisfying $y_i f(\boldsymbol{x}_i) = 1$, called support vectors
  - The solution does not change even data points that are not support vectors are removed

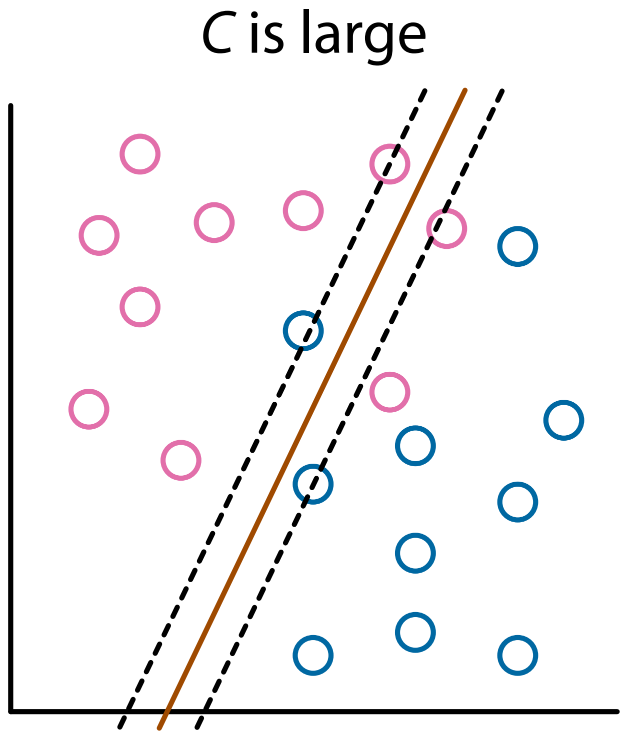
Margin

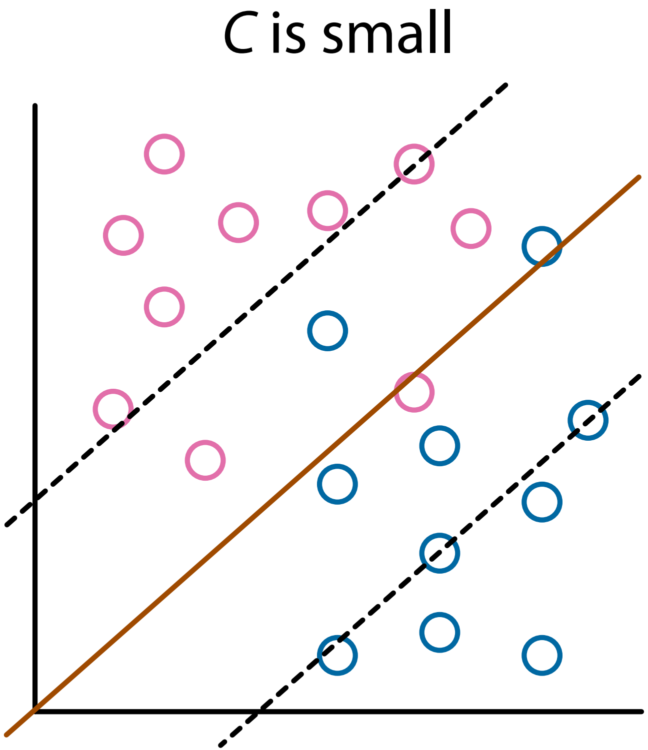

$f(\boldsymbol{x}) = \langle \boldsymbol{w}, x \rangle + w_0 = 0$
Margin
Support vector

Soft Margin

- Datasets are not often separable
- Extend SV classification to soft margin by relaxing $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + w_0 \geq 1$
- Change the constraint $y_i f(\boldsymbol{x}_i) \geq 1$ using the slack variable ξ_i to
  $y_i f(\boldsymbol{x}_i) = y_i (\langle \boldsymbol{w}, \boldsymbol{x} \rangle + w_0) \geq 1 - \xi_i, \quad i \in \{1, 2, \ldots, n\}$
- The formulation of soft margin SVM (C-SVM) is
  $$\min_{\boldsymbol{w}, w_0, \boldsymbol{\xi}} \frac{1}{2} ||\boldsymbol{w}||^2 + C \sum_{i \in \{1,2,\ldots,N\}} \xi_i \quad \text{s.t. } y_i f(\boldsymbol{x}_i) \geq 1 - \xi_i, \xi_i \geq 0, i \in \{1, 2, \ldots, N\}$$
  - C is called the regularization parameter

Soft Margin





Data Point Location

- $y_i f(\boldsymbol{x}_i) > 1$: $\boldsymbol{x}_i$ is outside margin
  - These points do not affect to the classification hyperplane
- $y_i f(\boldsymbol{x}_i) = 1$: $\boldsymbol{x}_i$ is on margin
- $y_i f(\boldsymbol{x}_i) < 1$: $\boldsymbol{x}_i$ is inside margin
  - These points do not exist in hard margin
- Points on margin and inside margin are support vectors

Dual Problem (1/4)

- The formulation of C-SVM

$$\min_{\boldsymbol{w}, w_0, \boldsymbol{\xi}} \frac{1}{2}||\boldsymbol{w}||^2 + C \sum_{i \in \{1,2,\ldots,N\}} \xi_i \quad \text{s.t. } y_i f(\boldsymbol{x}_i) \geq 1 - \xi_i, \xi_i \geq 0, i \in \{1, 2, \ldots, N\}$$

  is called the primal problem

- This is usually solved via the dual problem

- Make the Lagrange function using $\boldsymbol{a} = (a_1, \ldots, a_N), \boldsymbol{\mu} = (\mu_1, \ldots, \mu_N)$:

$$L(\boldsymbol{w}, w_0, \boldsymbol{\xi}, \boldsymbol{a}, \boldsymbol{\mu}) = \frac{1}{2}||\boldsymbol{w}||^2 + C \sum_{i \in [N]} \xi_i - \sum_{i \in [N]} a_i \big(y_i f(\boldsymbol{x}_i) - 1 + \xi_i\big) - \sum_{i \in [N]} \mu_i \xi_i$$

  - $[N] = \{1, 2, \ldots, N\}$

Dual Problem (2/4)

- Let us consider

$$D(\boldsymbol{\alpha}, \boldsymbol{\mu}) = \min_{\boldsymbol{w}, w_0, \boldsymbol{\xi}} L(\boldsymbol{w}, w_0, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu})$$

and its maximization

$$\max_{\boldsymbol{\alpha} \geq 0, \boldsymbol{\mu} \geq 0} D(\boldsymbol{\alpha}, \boldsymbol{\mu}) = \max_{\boldsymbol{\alpha} \geq 0, \boldsymbol{\mu} \geq 0} \min_{\boldsymbol{w}, w_0, \boldsymbol{\xi}} L(\boldsymbol{w}, w_0, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu})$$

- The inside minimization is achieved when

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i \in [N]} \alpha_i y_i \boldsymbol{x}_i = 0, \quad \frac{\partial L}{\partial w_0} = -\sum_{i \in [N]} \alpha_i y_i = 0, \quad \frac{\partial L}{\partial \xi_i} = C - \alpha_i - \mu_i = 0$$

Dual Problem (3/4)

- Putting the three conditions to the Lagrange function to remove $\boldsymbol{w}$, w_0, and $\boldsymbol{\xi}$, yielding

$$
\begin{aligned}
L &= \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i\in[N]}\xi_i - \sum_{i\in[N]}a_i\big(y_i f(\boldsymbol{x}_i) - 1 + \xi_i\big) - \sum_{i\in[N]}\mu_i\xi_i \\
&= \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i\in[N]}a_i y_i\langle\boldsymbol{w}, \boldsymbol{x}_i\rangle - w_0\sum_{i\in[N]}a_i y_i + \sum_{i\in[N]}a_i + \sum_{i\in[N]}(C - a_i - \mu_i)\xi_i \\
&= -\frac{1}{2}\sum_{i,j\in[N]}a_i a_j y_i y_j\langle\boldsymbol{x}_i, \boldsymbol{x}_j\rangle + \sum_{i\in[N]}a_i
\end{aligned}
$$

Dual Problem (4/4)

- It can be proved that $\max_{\boldsymbol{\alpha}\geq 0,\boldsymbol{\mu}\geq 0} \min_{\boldsymbol{w},w_0,\boldsymbol{\xi}} L(\boldsymbol{w}, w_0, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu})$, that is, the dual problem

$$\max_{\boldsymbol{\alpha}} -\frac{1}{2}\sum_{i,j\in[N]} \alpha_i\alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle + \sum_{i\in[N]} \alpha_i \quad \text{s.t.} \sum_{i\in[N]} \alpha_i y_i = 0,\ 0 \leq \alpha_i \leq C, i \in [N]$$

is equivalent to the primal problem

$$\min_{\boldsymbol{w},w_0,\boldsymbol{\xi}} \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i\in\{1,2,\ldots,N\}} \xi_i \quad \text{s.t. } y_i f(\boldsymbol{x}_i) \geq 1 - \xi_i, \xi_i \geq 0, i \in [N]$$

KKT (Karush-Kuhn-Tucker) condition

- The necessary conditions for a solution to be optimal:

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i \in [N]} \alpha_i y_i \boldsymbol{x}_i = 0, \ \frac{\partial L}{\partial w_0} = -\sum_{i \in [N]} \alpha_i y_i = 0, \ \frac{\partial L}{\partial \xi_i} = C - \alpha_i - \mu_i = 0$$

$$-\left(y_i f(\boldsymbol{x}_i) - 1 + \xi_i\right) \leq 0, \ -xi_i \leq 0,$$

$$\alpha_i \geq 0, \ \mu_i \geq 0,$$

$$\alpha_i\left(y_i f(\boldsymbol{x}_i) - 1 - \xi_i\right) = 0, \ \mu_i \xi_i = 0,$$

$$i \in [N]$$

Recovering Primal Variables

- Using these conditions, from the optimal $\boldsymbol{\alpha}$, we have

$$f(\boldsymbol{x}) = \sum_{i \in [N]} \alpha_i y_i \langle \boldsymbol{x}_i, \boldsymbol{x} \rangle + w_0,$$

$$w_0 = y_i - \sum_{j \in [N]} \alpha_j y_j \langle \boldsymbol{x}_j, \boldsymbol{x}_i \rangle, \quad \forall i \in \{i \in [N] \mid 0 < \alpha_i < C\}$$

  - Since the second condition holds for all $i \in \{i \in [N] \mid 0 < \alpha_i < C\}$, one can take the average to avoid numerical errors

Data Point Location

- $y_i f(\boldsymbol{x}_i) > 1 \iff \alpha_i = 0$: $\boldsymbol{x}_i$ is outside margin
  - These points do not affect to the classification hyperplane
- $y_i f(\boldsymbol{x}_i) = 1 \iff 0 < \alpha_i < C$: $\boldsymbol{x}_i$ is on margin
- $y_i f(\boldsymbol{x}_i) < 1 \iff \alpha_i = C$: $\boldsymbol{x}_i$ is inside margin
  - These points do not exist in hard margin
- Points on margin and inside margin are support vectors

How to Solve?

- The (dual) problem:

  $$\max_{\boldsymbol{a}} -\frac{1}{2}\boldsymbol{a}^T Q \boldsymbol{a} + \boldsymbol{1}^T \boldsymbol{a} \quad \text{s.t. } \boldsymbol{y}^T \boldsymbol{a} = 0,\ 0 \le \boldsymbol{a} \le C1$$

  - $Q \in \mathbb{R}^{N \times N}$ is the matrix such that $q_{ij} = y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$

- Since analytical solution is not available, iterative approach for continuous optimization with constraints is needed

- One of standard methods is the active set method

Active Set Method

- Divide the set $[N]$ of indices into three sets:

$$O = \{i \in [N] \mid a_i = 0\}$$

$$M = \{i \in [N] \mid 0 < a_i < C\}$$

$$I = \{i \in [N] \mid a_i = C\}$$

  - O and I are called active sets

- The problem can be solved using only $i \in M$, yielding

$$\begin{bmatrix} Q_M & \boldsymbol{y}_M \\ \boldsymbol{y}_M^T & 0 \end{bmatrix} \begin{bmatrix} a_M \\ v \end{bmatrix} = -C \begin{bmatrix} Q_{M,I} & 1 \\ 1^T & \boldsymbol{y}_I \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

  - This can be directly solved if Q_M is positive definite

Algorithm 2: Active Set Method

activeSetMethod(D)

  Initialize M, I, O

  while *there exists* i *s.t.* $y_i f(\boldsymbol{x}_i) < 1, i \in O$ *or* $y_i f(\boldsymbol{x}_i) > 1, i \in I$ **do**

    Update M, I, O

    repeat

      $\boldsymbol{\alpha}_M^{\text{new}} \leftarrow$ the solution of the above equation

      $\boldsymbol{d} \leftarrow \boldsymbol{\alpha}_M^{\text{new}} - \boldsymbol{\alpha}_M$

      $\boldsymbol{\alpha}_M \leftarrow \boldsymbol{\alpha}_M + \eta \boldsymbol{d}$; // the maximum η satisfying $\boldsymbol{\alpha}_M \in [0, C]^{|M|}$

      Move $i \in M$ from M to I or O if $\alpha_i = C$ or $\alpha_i = 0$

    until $\boldsymbol{\alpha}_M = \boldsymbol{\alpha}_M^{new}$;

Extension to Nonlinear Classification

- To achieve nonlinear classification, convert each data point $\boldsymbol{x}$ to some point $\varphi(\boldsymbol{x})$, and $f(\boldsymbol{x})$ becomes

  $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \varphi(\boldsymbol{x}) \rangle + w_0$

- The dual problem becomes

  $$\max_{\boldsymbol{\alpha}} -\frac{1}{2} \sum_{i,j \in [N]} \alpha_i \alpha_j y_i y_j \langle \varphi(\boldsymbol{x}_i), \varphi(\boldsymbol{x}_j) \rangle + \sum_{i \in [N]} \alpha_i \ \text{ s.t. } \sum_{i \in [N]} \alpha_i y_i = 0,\ 0 \le \alpha_i \le C, i \in [N]$$

  - Only the dot product $\langle \varphi(\boldsymbol{x}_i), \varphi(\boldsymbol{x}_j) \rangle$ is used!
  - We do not even need to know $\varphi(\boldsymbol{x}_i)$ and $\varphi(\boldsymbol{x}_j)$

- Kernel function: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \varphi(\boldsymbol{x}_i), \varphi(\boldsymbol{x}_j) \rangle$

C-SVM with Kernel Trick

- Using the kernel function K, we have

$$\max_{\boldsymbol{a}} -\frac{1}{2}\sum_{i,j\in[N]} a_i a_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_{i\in[N]} a_i \ \text{ s.t. } \sum_{i\in[N]} a_i y_i = 0,\ 0 \le a_i \le C, i \in [N]$$

  - The technique of using K is called kernel trick

Positive Definite Kernel

- A kernel $K : \Omega \times \Omega \to \mathbb{R}$ is a positive definite kernel if
  (i) $K(x, y) = K(y, x)$
  (ii) For $x_1, x_2, \ldots, x_N$, the $N \times N$ matrix

$$(K_{ij}) = \begin{bmatrix} K(x_1, x_1) & K(x_2, x_1) & \ldots & K(x_N, x_1) \\ K(x_1, x_2) & K(x_2, x_2) & \ldots & K(x_N, x_2) \\ \ldots & \ldots & \ldots & \ldots \\ K(x_1, x_N) & K(x_2, x_N) & \ldots & K(x_N, x_N) \end{bmatrix}$$

  is positive (semi-)definite, that is, $\sum_{i,j=1}^{N} c_i c_j K(x_i, x_j) \geq 0$ for any $c_1, c_2, \ldots, c_N \in \mathbb{R}$
  - $(K_{ij}) \in \mathbb{R}^{N \times N}$ is called the Gram matrix

Popular Positive Definite Kernels

- Linear Kernel

  $$K(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{x}, \boldsymbol{y} \rangle$$

- Gaussian (RBF) kernel

  $$K(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{1}{\sigma^2} ||\boldsymbol{x} - \boldsymbol{y}||^2\right)$$

- Polynomial Kernel

  $$K(\boldsymbol{x}, \boldsymbol{y}) = (\langle \boldsymbol{x}, \boldsymbol{y} \rangle + c)^c \qquad c, d \in \mathbb{R}$$

Simple Kernels

- The all-ones kernel

  $$K(\boldsymbol{x}, \boldsymbol{y}) = 1$$

- The delta (Dirac) kernel

  $$K(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} 1 & \text{if } \boldsymbol{x} = \boldsymbol{y}, \\ 0 & \text{otherwise} \end{cases}$$

Closure Properties of Kernels

- For two kernels K_1 and K_2, $K_1 + K_2$ is a kernel
- For two kernels K_1 and K_2, the product $K_1 \cdot K_2$ is a kernel
- For a kernel K and a positive scalar $\lambda \in \mathbb{R}^+$, λK is a kernel
- For a kernel K on a set D, its zero-extension:

  $$K_0(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} K(\boldsymbol{x}, \boldsymbol{y}) & \text{if } \boldsymbol{x}, \boldsymbol{y} \in D, \\ 0 & \text{otherwise} \end{cases}$$

  is a kernel

Kernels on Structured Data

- Given objects X and Y, decompose them into substructures S and T
- The R-convolution kernel K_R by Haussler (1999) is given as

$$K_R(X, Y) = \sum_{s \in S, t \in T} K_{\text{base}}(s, t)$$

  - K_{base} is an arbitrary base kernel, often the delta kernel
- For example, X is a graph and S is the set of all subgraphs

Summary

- SVM finds the “best” classification hyperplane
  - The margin is maximized
- Although the original SVM can perform only linear classification, it can be extended to nonlinear classification by using kernels
- Gaussian kenrel + C-SVM can be the first choice