Introduction to Machine Learning

Kernel Methods

SHENG-JYH WANG

NATIONAL YANG MING CHIAO TUNG UNIVERSITY, TAIWAN

SPRING, 2024

Prerequisite Knowledge

Lagrange Multipliers (1/8)

Used to find the stationary points of a function of several variables subject to one or more constraints.

Consider a *D*-dimensional variable **x** with components $x_1, \ldots, x_D$.

- The constraint equation $g(\mathbf{x}) = 0$ then represents a (D-1)-dimensional surface in **x**-space.

At any point on the constraint surface, the gradient $\nabla g(x)$ of the constraint function will be orthogonal to the surface.

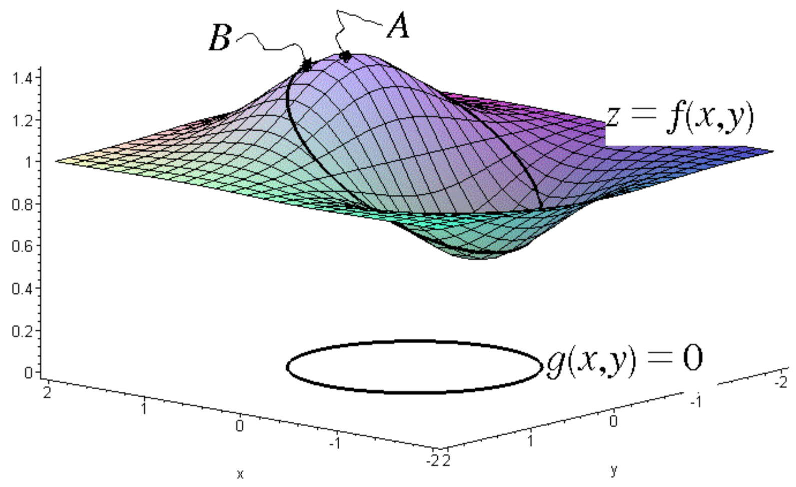
$$g(\mathbf{x} + \boldsymbol{\epsilon}) \simeq g(\mathbf{x}) + \boldsymbol{\epsilon}^{\mathrm{T}} \nabla g(\mathbf{x}).$$

If $g(\mathbf{x}) = g(\mathbf{x}+\varepsilon)$ and $\varepsilon \rightarrow 0$, $\varepsilon^{\mathrm{T}} \nabla g(\mathbf{x}) = 0$.

Lagrange Multipliers (2/8)

Equality Constraint

If we seek a point **x*** on the constraint surface $g(\mathbf{x}) = 0$ such that $f(\mathbf{x})$ is maximized. The vector $\nabla f(\mathbf{x})$ will be orthogonal to the constraint surface at x*.



(Reference: http://staff.www.ltu.se/~tomas/applmath/chap7en/part7.html)

Lagrange Multipliers (3/8)

$$\nabla f + \lambda \nabla g = 0$$ where $\lambda \neq 0$.

We can define the *Lagrangian* function:

$$L(\mathbf{x}, \lambda) \equiv f(\mathbf{x}) + \lambda g(\mathbf{x}).$$

$$\nabla_{\mathbf{x}} L = 0$$

$$\partial L / \partial \lambda = 0$$

Lagrange Multipliers (4/8)

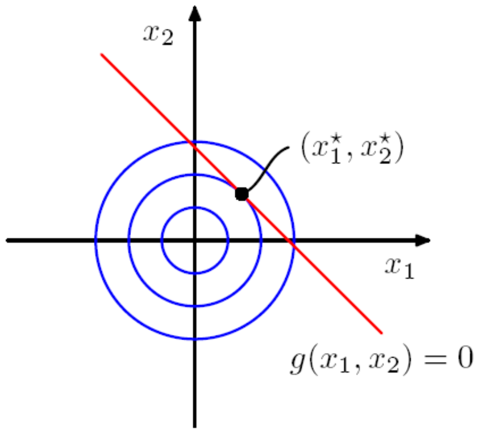
Example: $f(x_1, x_2) = 1 - x_1^2 - x_2^2$

subject to the constraint $g(x_1, x_2) = x_1 + x_2 - 1 = 0$

$$L(\mathbf{x}, \lambda) = 1 - x_1^2 - x_2^2 + \lambda(x_1 + x_2 - 1)$$

$$\begin{aligned} -2x_1 + \lambda &= 0 \\ -2x_2 + \lambda &= 0 \\ x_1 + x_2 - 1 &= 0. \end{aligned}$$

$$(x_1^\star, x_2^\star) = (\tfrac{1}{2}, \tfrac{1}{2}) \qquad \lambda = 1$$



Lagrange Multipliers (5/8)

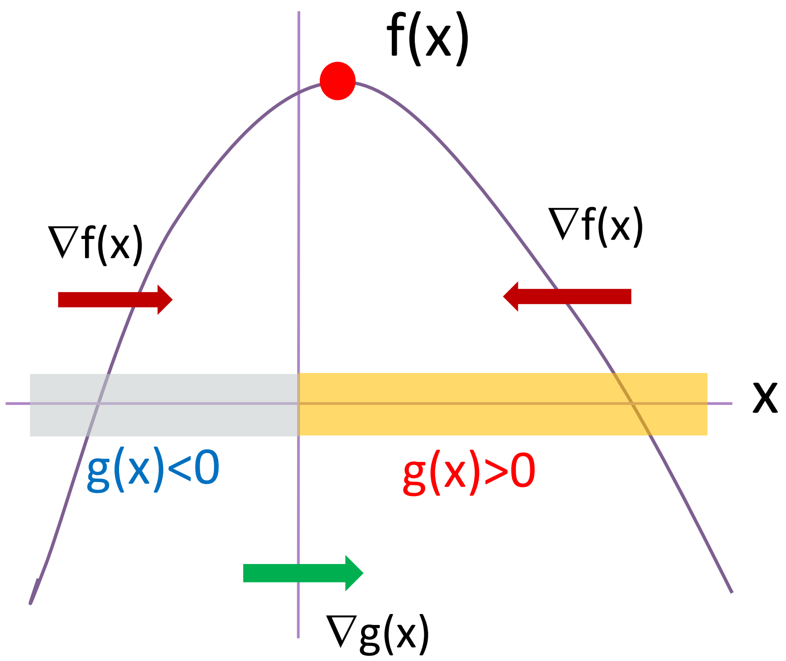
Inequality Constraint

We consider the problem of maximizing $f(\mathbf{x})$ subject to an *inequality constraint* of the form $g(\mathbf{x}) \geq 0$.

- ✓ If the constrained stationary point lies in the region where $g(\mathbf{x}) > 0$, the constraint is said to be *inactive.* The function $g(\mathbf{x})$ plays no role and the stationary condition is simply $\nabla f(\mathbf{x}) = 0$. ($\lambda = 0$)
- ✓ If the constrained stationary point lies on the boundary $g(\mathbf{x}) = 0$, the constraint is said to be *active*. The solution lies on the boundary, is analogous to the equality constraint. ($\lambda > 0$)

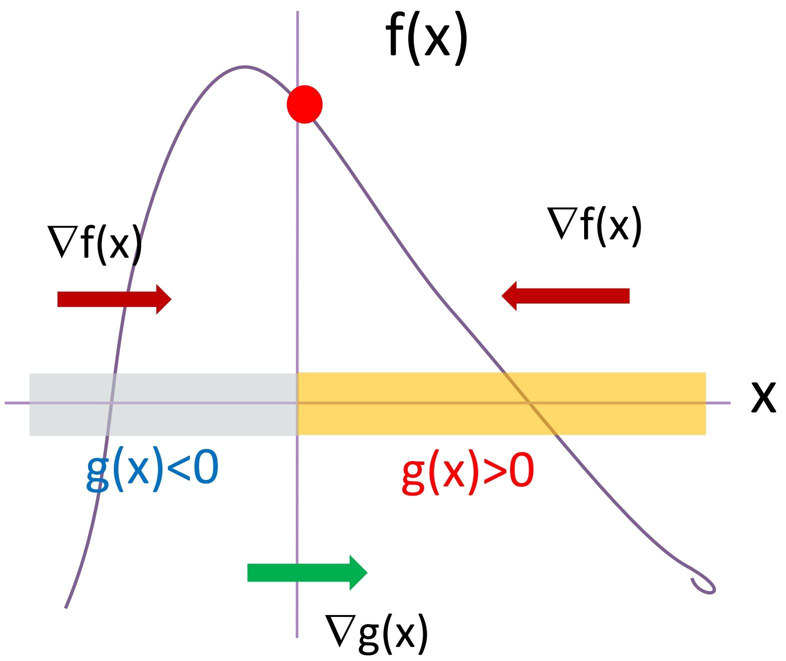
Lagrange Multipliers (6/8)

Inactive Constraint



$$\nabla f = \lambda \nabla g$$

Active Constraint



$$\nabla f = -\lambda \nabla g$$

Gradient 方向相反
最大值在Gradient =0

Lagrange Multipliers (7/8)

The solution to problem of maximizing $f(\mathbf{x})$ subject to $g(\mathbf{x}) \geq 0$ is obtained by optimizing the Lagrange function $L(\mathbf{x},\lambda) \equiv f(\mathbf{x}) + \lambda g(\mathbf{x})$ with respect to **x** and λ subject to the conditions

$$\begin{aligned} g(\mathbf{x}) &\geqslant 0 \\ \lambda &\geqslant 0 \\ \lambda g(\mathbf{x}) &= 0 \end{aligned}$$

Karush-Kuhn-Tucker (KKT) conditions

Remark: If we wish to minimize the function $f(\mathbf{x})$ subject to an inequality constraint $g(\mathbf{x}) \geq 0$, then we minimize the Lagrangian function $L(\mathbf{x},\lambda) \equiv f(\mathbf{x}) - \lambda g(\mathbf{x})$ with respect to **x**, subject to $\lambda \geq 0$.

Lagrange Multipliers (8/8)

Suppose we wish to maximize $f(\mathbf{x})$ subject to $g_j(\mathbf{x}) = 0$ for $j = 1, \ldots, J$, and $h_k(\mathbf{x}) \geq 0$ for $k = 1, \ldots, K$. We introduce Lagrange multipliers $\{\lambda_j\}$ and $\{\mu_k\}$, and optimize the Lagrangian function given by

$$L(\mathbf{x}, \{\lambda_j\}, \{\mu_k\}) = f(\mathbf{x}) + \sum_{j=1}^{J} \lambda_j g_j(\mathbf{x}) + \sum_{k=1}^{K} \mu_k h_k(\mathbf{x})$$

subject to $\mu_k \geq 0$ and $\mu_k h_k(\mathbf{x}) = 0$ for $k = 1, \ldots, K$.

Kernel Methods

Introduction (1/5)

Recall the linear models for regression

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n), \beta^{-1}) = N(\mathbf{t} \mid \mathbf{\Phi w}, \beta^{-1}\mathbf{I})$$ Likelihood Function

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$ Prior

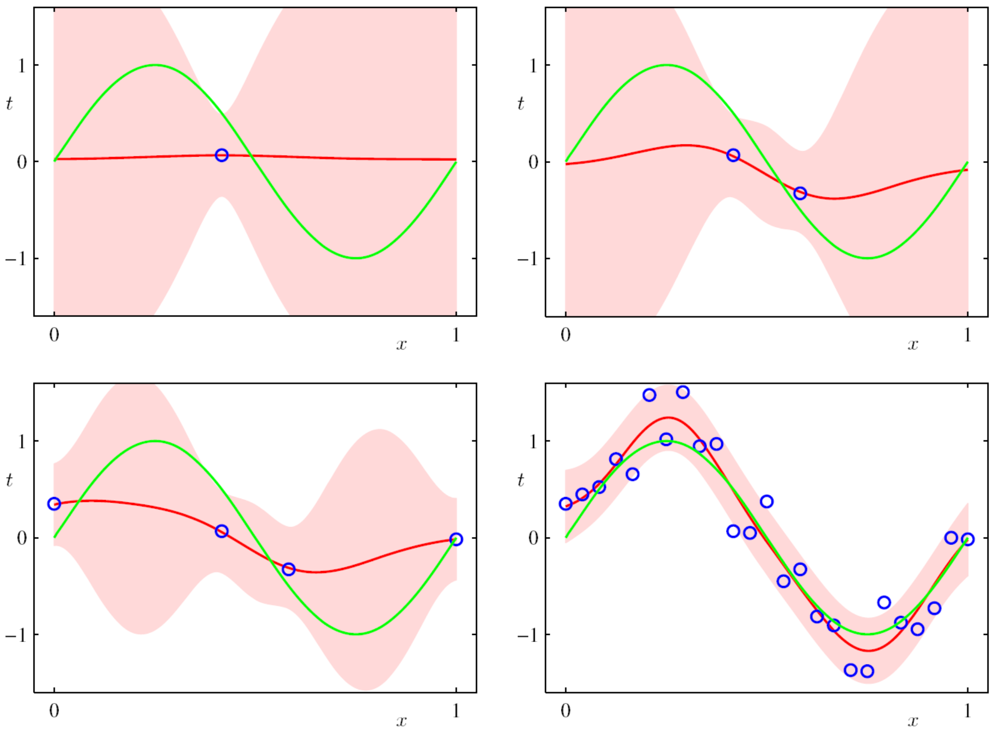
$$p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$$ Posterior

where
$$\begin{aligned} \mathbf{m}_N &= \mathbf{S}_N \left(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\mathbf{\Phi}^{\mathrm{T}}\mathbf{t}\right) \\ \mathbf{S}_N^{-1} &= \mathbf{S}_0^{-1} + \beta\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}. \end{aligned}$$

$$\Rightarrow \quad p(t|\mathbf{x}, \mathbf{t}, \alpha, \beta) = \mathcal{N}(t|\mathbf{m}_N^{\mathrm{T}}\phi(\mathbf{x}), \sigma_N^2(\mathbf{x}))$$ Predictive Distribution

where $$\sigma_N^2(\mathbf{x}) = \frac{1}{\beta} + \phi(\mathbf{x})^{\mathrm{T}}\mathbf{S}_N\phi(\mathbf{x})$$

Introduction (2/5)



red curve: mean of the predictive distribution
red shaded region: one standard deviation span around the mean

Introduction (3/5)

Equivalent Kernel

The predictive mean can be written as

$$y(\mathbf{x}, \mathbf{m}_N) = \mathbf{m}_N^{\mathrm{T}} \phi(\mathbf{x}) = \beta \phi(\mathbf{x})^{\mathrm{T}} \mathbf{S}_N \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{t} = \sum_{n=1}^{N} \beta \phi(\mathbf{x})^{\mathrm{T}} \mathbf{S}_N \phi(\mathbf{x}_n) t_n$$

where $\mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \beta \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi}$

$$\Rightarrow \quad y(\mathbf{x}, \mathbf{m}_N) = \sum_{n=1}^{N} k(\mathbf{x}, \mathbf{x}_n) t_n$$

where $k(\mathbf{x}, \mathbf{x}') = \beta \phi(\mathbf{x})^{\mathrm{T}} \mathbf{S}_N \phi(\mathbf{x}')$

*the **smoother matrix** (or the **equivalent kernel**)*

The prediction at x is given by a linear combination of the target values from the training set!

Introduction (4/5)

Remarks:

1. Instead of introducing a set of basis functions, we can define a localized kernel directly and use it to make predictions for new input vectors x, given the observed training set.
2. It can be shown that

$$\boxed{\sum_{n=1}^{N} k(\mathbf{x}, \mathbf{x}_n) = 1}$$

for all values of **x**.

Introduction (5/5)

- The kernel concept was introduced into the field of pattern recognition by Aizerman *et al.* (1964)
- Re-introduced into machine learning in the context of large-margin classifiers by Boser *et al.* (1992)
- The concept of a kernel formulated as an inner product in a feature space allows us to build interesting extensions of many well-known algorithms by making use of the *kernel trick*, also known as *kernel substitution*.

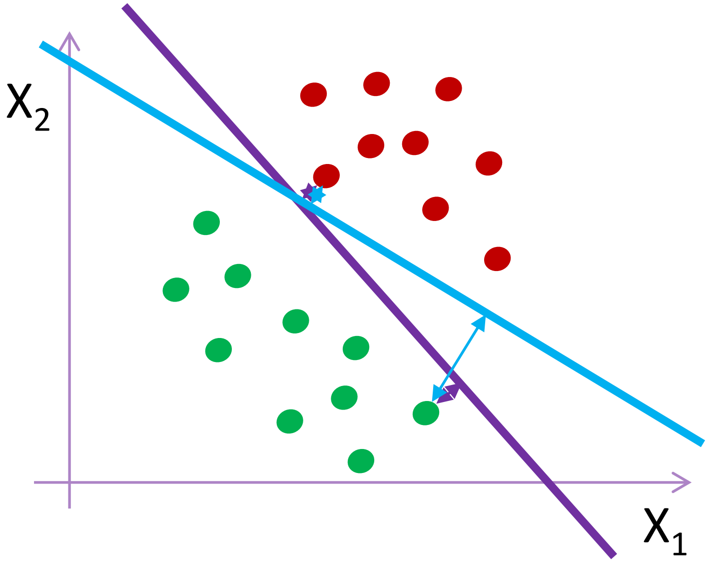
Sparse Kernel Machine

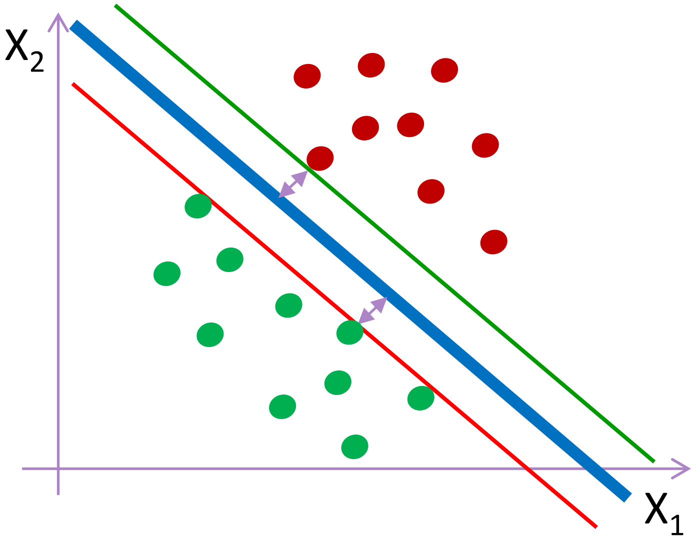
Look for kernel-based algorithms whose predictions for new inputs depend only on the kernel function evaluated at a subset of the training data points.

- ✓ *Support Vector Machine* (SVM)
- ✓ *Relevance Vector Machine* (RVM)

Concept of Maximum Margin Classifier

Which cutting line is better?



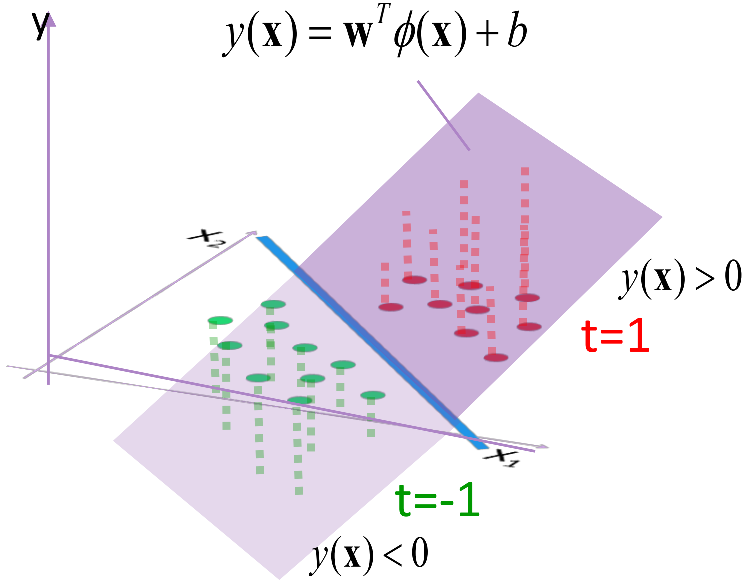


Here, we aim to choose decision boundary for which the *margin* is maximized.

Support Vector Machine (1/24)

Training data: N input vectors $x_1, \ldots, x_N$, with corresponding target values $t_1, \ldots, t_N$ where $t_n \in \{-1, 1\}$.

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}) + b$$



Linearly separable training data

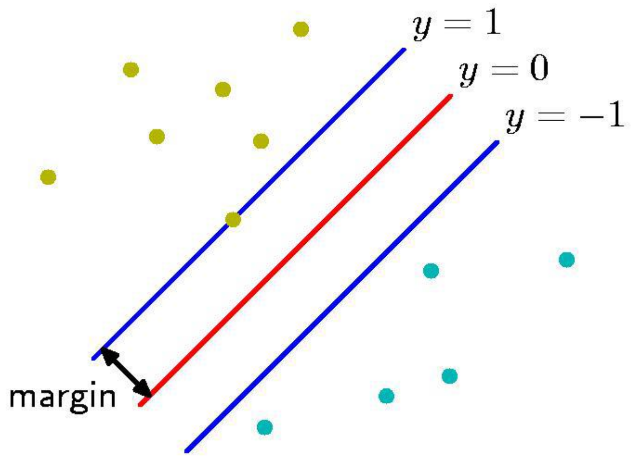
There exists at least one choice of the parameters $\boldsymbol{w}$ and b such that

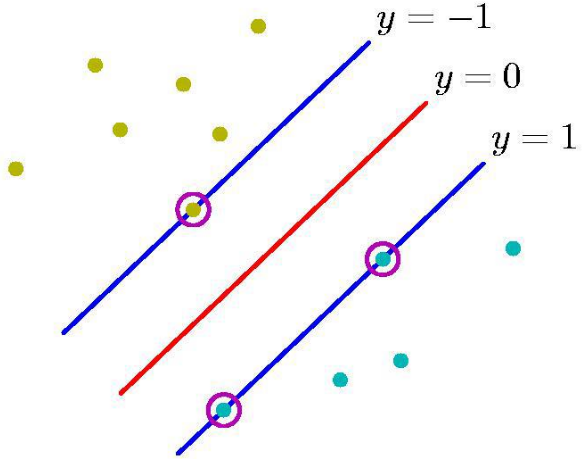
$y(\mathbf{x}_n) > 0$ for $t_n = +1$ and $y(\mathbf{x}_n) < 0$ for $t_n = -1$.

Therefore, $t_n y(\mathbf{x}_n) > 0$ for all *training* points.

Support Vector Machine (2/24)

Margin: the smallest distance between the decision boundary and any of the samples.



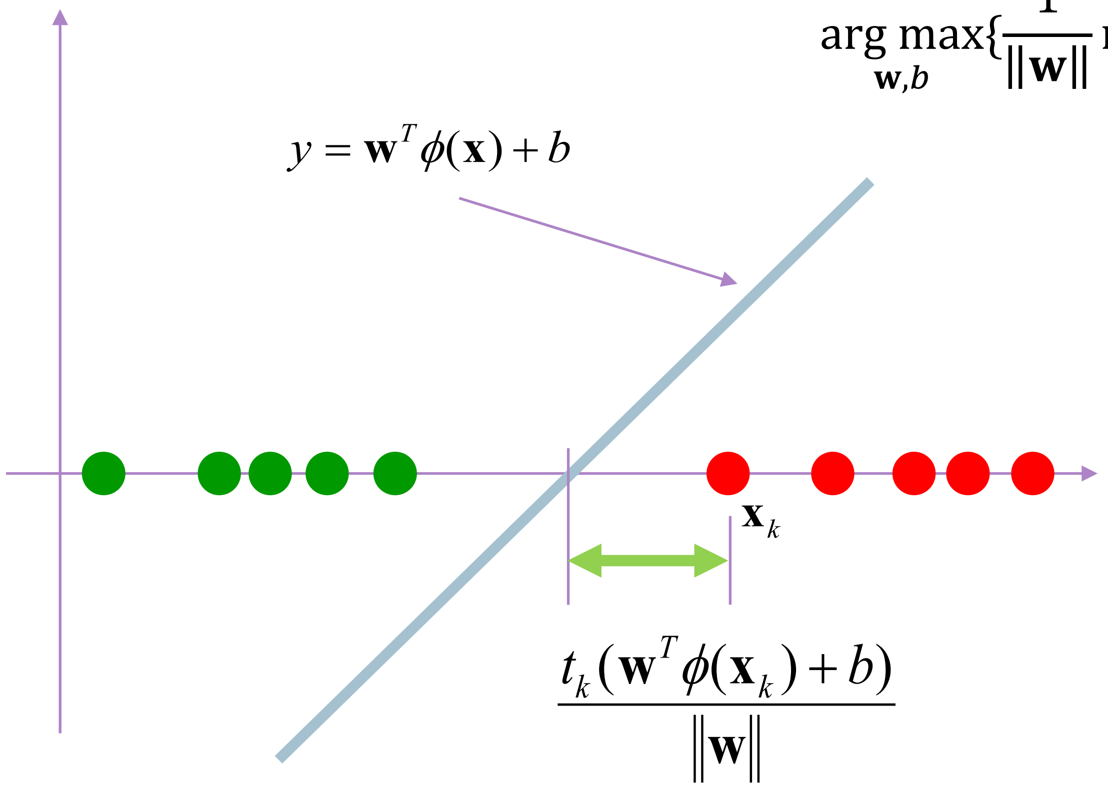


The distance of a point x_n to the decision surface:

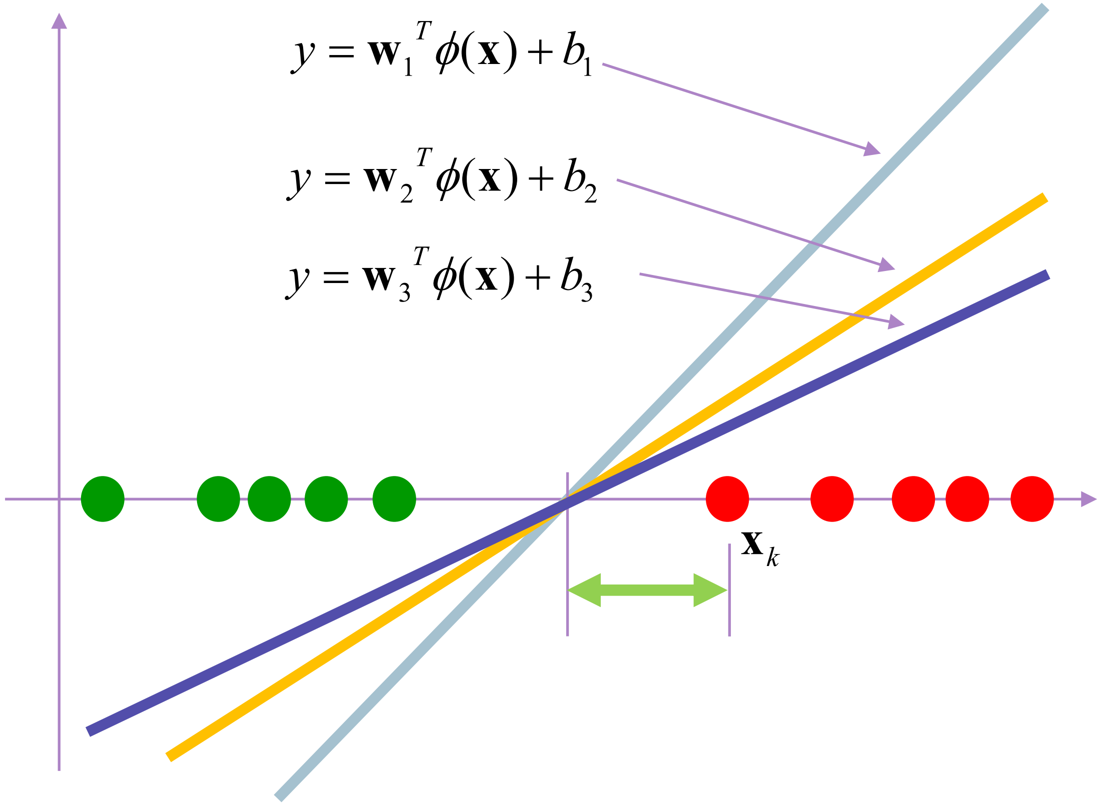
$$\frac{t_n y(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{t_n(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) + b)}{\|\mathbf{w}\|}$$

Support Vector Machine (3/24)

$$\arg\max_{\mathbf{w},b}\{\frac{1}{\|\mathbf{w}\|}\min_{n}\{t_n(\mathbf{w}^T\phi(\mathbf{x}_n)+b)\}\}$$



Support Vector Machine (4/24)



$$\frac{t_k(\mathbf{w}_1^T \phi(\mathbf{x}_k) + b_1)}{\|\mathbf{w}_1\|}$$

$$= \frac{t_k(\mathbf{w}_2^T \phi(\mathbf{x}_k) + b_2)}{\|\mathbf{w}_2\|}$$

$$= \frac{t_k(\mathbf{w}_3^T \phi(\mathbf{x}_k) + b_3)}{\|\mathbf{w}_3\|}$$

Support Vector Machine (5/24)

The maximum margin solution is found by solving

$$\underset{\mathbf{w},b}{\arg\max}\left\{\frac{1}{\|\mathbf{w}\|}\min_n\left[t_n\left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)+b\right)\right]\right\}$$

However, direct solution of this optimization problem is very complex!

Note that if $\mathbf{w} \to \kappa\mathbf{w}$ and $b \to \kappa b$, the distance from any point $\mathbf{x}_n$ to the decision surface is unchanged.

Hence, we set

$$t_n\left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)+b\right)=1$$

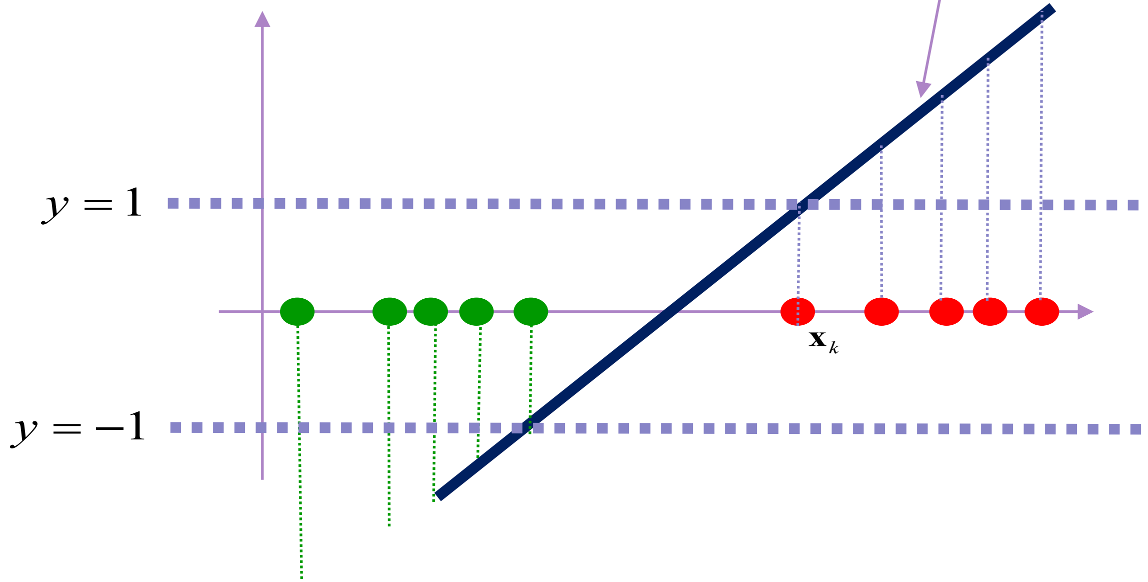
for the point that is closest to the surface.

Support Vector Machine (6/24)

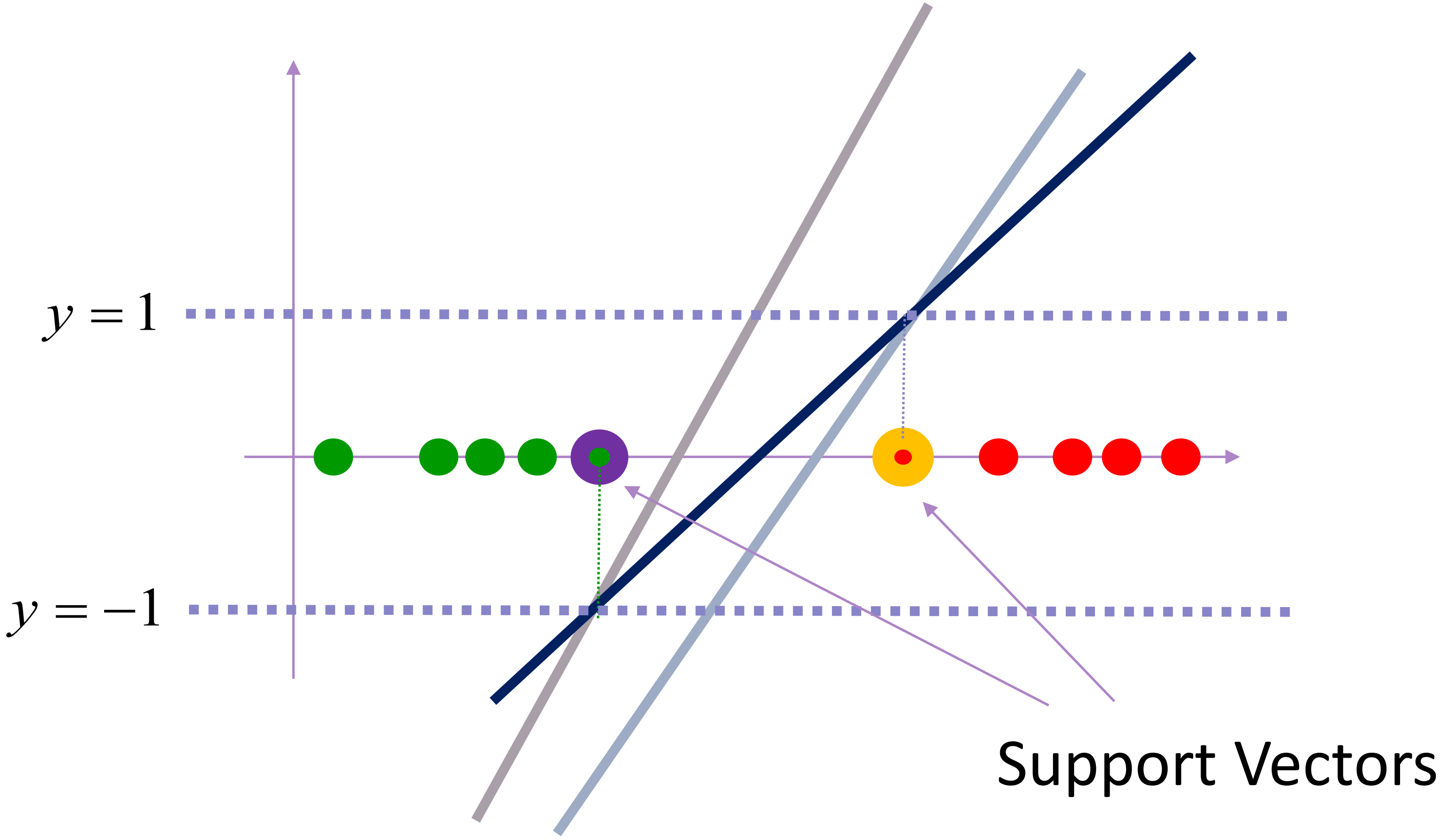
$$\arg\max_{\mathbf{w},b}\{\frac{1}{\|\mathbf{w}\|}\min_n\{t_n(\mathbf{w}^T\phi(\mathbf{x}_n)+b)\}\}$$

$$\underset{\mathbf{w},b}{\operatorname{argmin}}\frac{1}{2}\|\mathbf{w}\|^2 \quad \text{subject to} \;\; t_n(w^T\varphi(x_n)+b)\geq 1$$

$$y=\mathbf{w}^T\phi(\mathbf{x})+b$$



Support Vector Machine (7/24)



Support Vector Machine (8/24)

$$\Rightarrow \qquad t_n \left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) + b\right) \geqslant 1, \qquad\qquad n = 1, \ldots, N.$$

The canonical representation of the decision hyperplane.

The optimization problem now becomes

$$\underset{\mathbf{w},b}{\arg\min}\, \frac{1}{2}\|\mathbf{w}\|^2$$

subject to $$t_n \left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) + b\right) \geqslant 1, \qquad\qquad n = 1, \ldots, N.$$

a quadratic programming problem!

Support Vector Machine (9/24)

Minimize $L(\mathbf{w}, b, a) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n \{t_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) - 1\}$

KKT Conditions

$$a_n \geq 0$$

$$t_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) - 1 \geq 0$$

$$a_n \{t_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) - 1\} = 0$$

For those $t_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) > 1$, we have $a_n = 0$

Support Vector Machine (10/24)

$$L(\mathbf{w}, b, a) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n \{t_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) - 1\}$$

Setting $\dfrac{\partial L(\mathbf{w}, b, a)}{\partial \mathbf{w}} = 0 \qquad \dfrac{\partial L(\mathbf{w}, b, a)}{\partial b} = 0$

$$\Rightarrow \quad \mathbf{w} = \sum_{n=1}^{N} a_n t_n \phi(\mathbf{x}_n) \qquad 0 = \sum_{n=1}^{N} a_n t_n$$

Support Vector Machine (11/24)

By eliminating **w** and *b* from $L(\mathbf{w}, b, \mathbf{a})$, we get the *dual representation* of the maximum margin problem in which we maximize

$$\boxed{\widetilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m)}$$

subject to

$$a_n \geqslant 0, \qquad n = 1, \ldots, N,$$

$$\sum_{n=1}^{N} a_n t_n = 0.$$

Kernel function: $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^{\mathrm{T}} \phi(\mathbf{x}')$

Support Vector Machine (12/24)

To classify new data points using the trained model, we evaluate the sign of $y(\mathbf{x})$ defined by $y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}) + b$.

With $\mathbf{w} = \sum_{n=1}^{N} a_n t_n \phi(\mathbf{x}_n)$, we have

$$\boxed{y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b.}$$

Support Vector Machine (13/24)

Karush-Kuhn-Tucker (KKT) conditions:

$$
\begin{aligned}
a_n &\geqslant 0 \\
t_n y(\mathbf{x}_n) - 1 &\geqslant 0 \\
a_n \{t_n y(\mathbf{x}_n) - 1\} &= 0.
\end{aligned}
$$

For every data point, either $a_n = 0$ or $t_n y(\mathbf{x}_n) = 1$.

✓ $a_n = 0 \Rightarrow$ That data point plays no role in making predictions for new data points.

✓ $a_n \neq 0 \Rightarrow$ That data point is called **support vector** and lies on the maximum margin hyperplanes in feature space.

Support Vector Machine (14/24)

For any support vector $\mathbf{x}_n$, we have

$$\underset{\substack{\shortparallel \\ \pm 1}}{t_n} \left(\sum_{m \in \mathcal{S}} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) + b \right) = 1$$

$\Rightarrow$ The threshold b can be determined by calculating

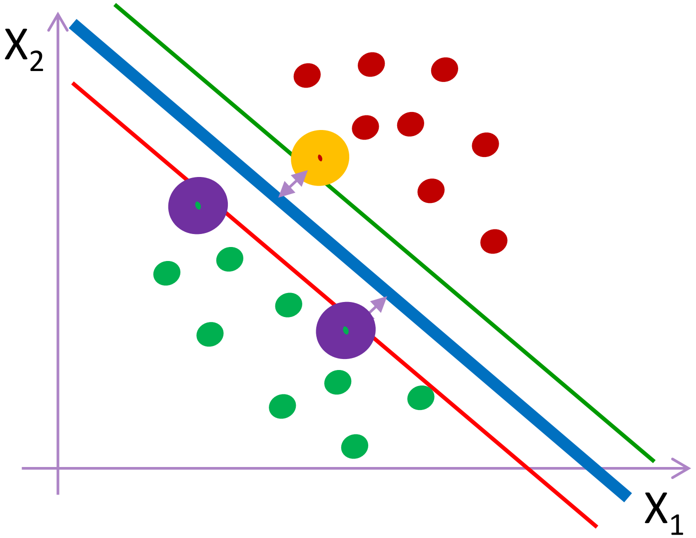
$$b = \frac{1}{N_{\mathcal{S}}} \sum_{n \in \mathcal{S}} \left(t_n - \sum_{m \in \mathcal{S}} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) \right)$$

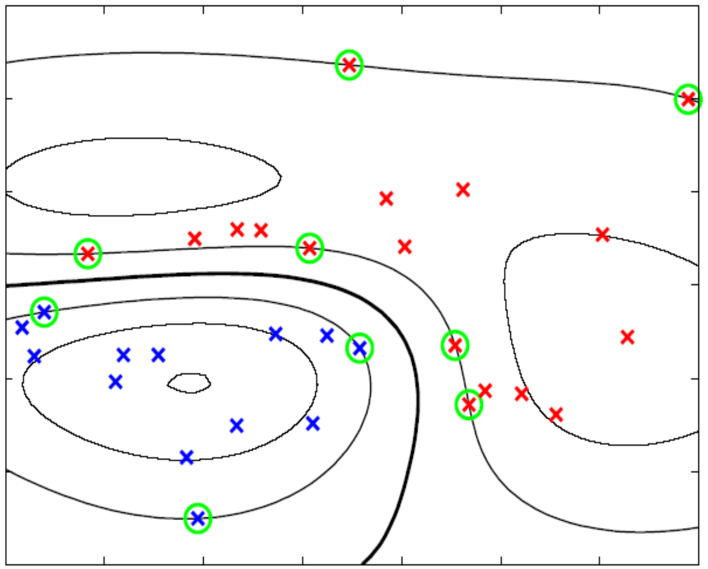
S: the set of indices of the support vectors

N_S: the total number of support vectors.

Support Vector Machine (15/24)

$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b$$





Ref: C.M. Bishop, Pattern Recognition & Machine Learning

Support Vector Machine (16/24)

The maximum margin classifier can also expressed as the minimization of an error function, with a simple quadratic regularizer:

$$\sum_{n=1}^{N} E_{\infty}(y(\mathbf{x}_n)t_n - 1) + \lambda\|\mathbf{w}\|^2$$

where $E_{\infty}(z)$ is a function that is zero if $z \geq 0$ and ∞ otherwise.

Support Vector Machine (17/24)

Overlapping Class Distributions

Allow data points to be on the "wrong side" of the margin boundary, but with a penalty that increases with the distance from that boundary.

$\Rightarrow$ Introduce *slack variables*, $\xi_n \geq 0$ where $n = 1, \ldots, N$.

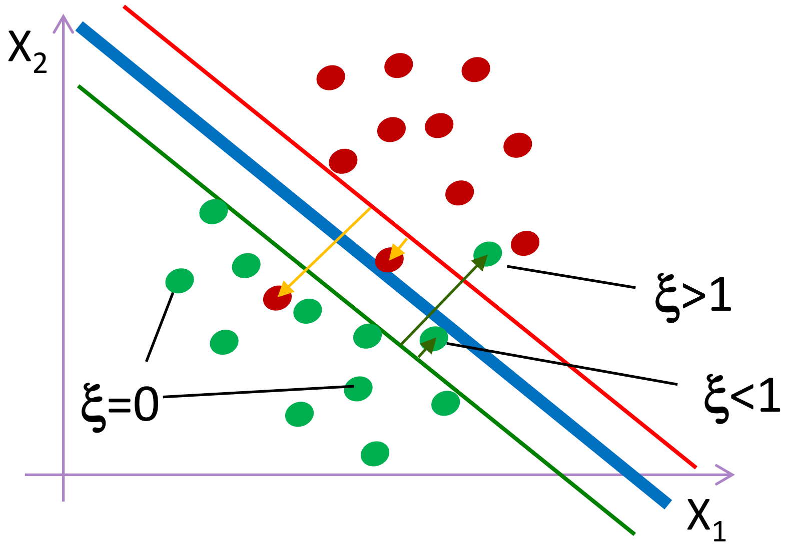
- ✓ $\xi_n = 0$ for data points on or inside the correct margin boundary.
- ✓ $\xi_n = |t_n - y(\mathbf{x}_n)|$ for other points.
  - Data points inside the margin, but on the correct side of the decision boundary. $\Rightarrow 0 < \xi_n < 1$.
  - Data points on the decision boundary $y(\mathbf{x}_n) = 0 \Rightarrow \xi_n = 1$.
  - Data points on the wrong side of the decision boundary. $\Rightarrow \xi_n > 1$.

Support Vector Machine (18/24)

Now, we have the classification constraints

$$t_n y(\mathbf{x}_n) \geqslant 1 - \xi_n, \qquad n = 1, \ldots, N$$

where $\xi_n \geq 0$.



Wish to minimize

$$C \sum_{n=1}^{N} \xi_n + \frac{1}{2}\|\mathbf{w}\|^2$$

subject to $t_n y(\mathbf{x}_n) \geqslant 1 - \xi_n, \qquad n = 1, \ldots, N$

and $\xi_n \geq 0$.

Support Vector Machine (19/24)

The Lagrangian is

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n - \sum_{n=1}^{N} a_n \{t_n y(\mathbf{x}_n) - 1 + \xi_n\} - \sum_{n=1}^{N}\mu_n\xi_n$$

where $\{a_n \geq 0\}$ and $\{\mu_n \geq 0\}$ are Lagrange multipliers.

KKT conditions

$$\begin{aligned} a_n &\geqslant 0 \\ t_n y(\mathbf{x}_n) - 1 + \xi_n &\geqslant 0 \\ a_n\left(t_n y(\mathbf{x}_n) - 1 + \xi_n\right) &= 0 \\ \mu_n &\geqslant 0 \\ \xi_n &\geqslant 0 \\ \mu_n\xi_n &= 0 \end{aligned}$$

where $n = 1, \ldots, N$.

Support Vector Machine (20/24)

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{n=1}^{N} a_n t_n \boldsymbol{\phi}(\mathbf{x}_n)$$

$$\frac{\partial L}{\partial b} = 0 \quad \Rightarrow \quad \sum_{n=1}^{N} a_n t_n = 0$$

$$\frac{\partial L}{\partial \xi_n} = 0 \quad \Rightarrow \quad a_n = C - \mu_n.$$

Support Vector Machine (21/24)

Dual representation

Maximizing

$$\widetilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m)$$

subject to

$$0 \leqslant a_n \leqslant C$$

$$\sum_{n=1}^{N} a_n t_n = 0$$

Support Vector Machine (22/24)

- ✓ $a_n = 0$: such data points do not contribute to the predictive model.
- ✓ $0 < a_n < C \Rightarrow \mu_n > 0 \Rightarrow \xi_n = 0 \Rightarrow$ such points lie on the margin.
- ✓ $a_n = C \Rightarrow$ such points lie inside the margin and can either be correctly classified if $\xi_n \leq 1$ or misclassified if $\xi_n > 1$.

$$b = \frac{1}{N_{\mathcal{M}}} \sum_{n \in \mathcal{M}} \left(t_n - \sum_{m \in \mathcal{S}} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) \right)$$

M: the set of indices of data points having $0 < a_n < C$.

S: the set of indices of the support vectors

Support Vector Machine (23/24)

ν-***SVM**:* An alternative, equivalent formulation of the SVM

Maximizing $$\widetilde{L}(\mathbf{a}) = -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m)$$

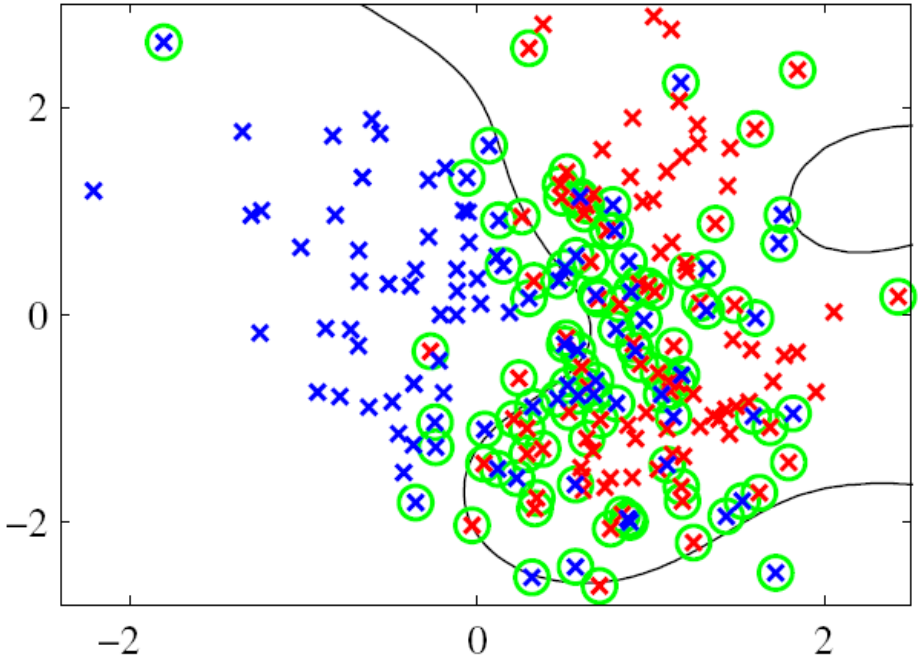
subject to $$0 \leqslant a_n \leqslant 1/N$$

$$\sum_{n=1}^{N} a_n t_n = 0$$

$$\sum_{n=1}^{N} a_n \geqslant \nu.$$

Support Vector Machine (24/24)

Illustration of the ν-SVM applied to a nonseparable data set in two dimensions. The support vectors are indicated by circles.



Limitations of SVM

- ✓ The outputs of an SVM represent decisions rather than posterior probabilities.
- ✓ The SVM was originally formulated for two classes, and the extension to $K > 2$ classes is problematic.
- ✓ The complexity parameter C, or ν (and ε in the case of regression), must be found using a hold-out method, such as cross-validation.
- ✓ The kernel functions are required to be positive definite.