Margin optimization

\[ \min \frac{1}{2} \|w\|_2^2 \]
subject to \( y_i(\langle w_0, w \rangle + w^T x_i) - 1 \geq 0 \), \( i = 1, 2, \ldots, n \).

What are support vectors?
Margin optimization

\[
\min_{(w, w_0)} \frac{1}{2} \|w\|^2 \\
\text{subject to } y_i(w_0 + w^T x_i) - 1 \geq 0, \quad i = 1, \ldots, n.
\]
Margin optimization

\[
\min_{(w, w_0)} \frac{1}{2} \|w\|^2
\]

subject to \( y_i(w_0 + w^T x_i) - 1 \geq 0, \quad i = 1, \ldots, n. \)

What are support vectors?
Margin optimization

\[
\min_{(w, w_0)} \frac{1}{2} \|w\|^2
\]

subject to \(y_i(w_0 + w^T x_i) - 1 \geq 0, \ i = 1, \ldots, n.\)

What are support vectors?
Margin optimization

\[
\begin{align*}
\min_{(w, w_0)} & \quad \frac{1}{2} \|w\|^2 \\
\text{subject to} & \quad y_i (w_0 + w^T x_i) - 1 \geq 0, \quad i = 1, \ldots, n.
\end{align*}
\]
Margin optimization

\[
\min_{(w,w_0)} \frac{1}{2} \|w\|^2
\]

subject to \( y_i(w_0 + w^T x_i) - 1 \geq 0, \quad i = 1, \ldots, n. \)

Add constraints as terms to the objective function:

\[
\min_{(w,w_0)} \frac{1}{2} \|w\|^2 + \sum_{i=1}^{n} \max_{\alpha_i \geq 0} \alpha_i \left[ 1 - y_i(w_0 + w^T x_i) \right]
\]
Margin optimization

\[
\min_{(\mathbf{w},w_0)} \frac{1}{2} \| \mathbf{w} \|^2
\]

subject to \( y_i(w_0 + \mathbf{w}^T \mathbf{x}_i) - 1 \geq 0, \quad i = 1, \ldots, n. \)

Add constraints as terms to the objective function:

\[
\min_{(\mathbf{w},w_0)} \frac{1}{2} \| \mathbf{w} \|^2 + \sum_{i=1}^{n} \max_{\alpha_i \geq 0} \alpha_i \left[ 1 - y_i(w_0 + \mathbf{w}^T \mathbf{x}_i) \right]
\]

What if \( y_i(w_0 + \mathbf{w}^T \mathbf{x}_i) > 1 \) (not a support vector) at the solution?
Margin optimization

\[
\begin{align*}
\min_{(w, w_0)} & \quad \frac{1}{2} \|w\|^2 \\
\text{subject to} & \quad y_i(w_0 + w^T x_i) - 1 \geq 0, \quad i = 1, \ldots, n.
\end{align*}
\]

Add constraints as terms to the objective function:

\[
\begin{align*}
\min_{(w, w_0)} & \quad \frac{1}{2} \|w\|^2 + \sum_{i=1}^{n} \max_{\alpha_i \geq 0} \alpha_i \left[ 1 - y_i(w_0 + w^T x_i) \right]
\end{align*}
\]

What if \( y_i(w_0 + w^T x_i) > 1 \) (not a support vector) at the solution? Then we must have \( \alpha_i = 0 \).
Review: Support vector machines (separable case)

Margin optimization

\[
\min_{(w, w_0)} \frac{1}{2} \|w\|^2
\]

subject to \( y_i(w_0 + w^T x_i) - 1 \geq 0, \quad i = 1, \ldots, n. \)

Add constraints as terms to the objective function:

\[
\min_{(w, w_0)} \frac{1}{2} \|w\|^2 + \sum_{i=1}^{n} \max_{\alpha_i \geq 0} \alpha_i \left[ 1 - y_i(w_0 + w^T x_i) \right]
\]

What if \( y_i(w_0 + w^T x_i) > 1 \) (not a support vector) at the solution?
Then we must have \( \alpha_i = 0 \).
Review: SVM optimization (separable case)

\[
\max_{\{\alpha_i \geq 0\}} \min_{(\mathbf{w}, w_0)} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^{n} \alpha_i \left[ 1 - y_i (w_0 + \mathbf{w}^T \mathbf{x}_i) \right] \right\}
\]

\[ L(\mathbf{w}, w_0; \alpha) \]

First, we fix \(\alpha\) and minimize \(L(\mathbf{w}, w_0; \alpha)\) w.r.t. \(\mathbf{w}, w_0\):

\[
\frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, w_0; \alpha) = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0,
\]

\[
\frac{\partial}{\partial w_0} L(\mathbf{w}, w_0; \alpha) = -\sum_{i=1}^{n} \alpha_i y_i = 0.
\]
Review: SVM optimization (separable case)

\[
\text{max} \quad \text{min} \quad \left\{ \frac{1}{2} \| w \|^2 + \sum_{i=1}^{n} \alpha_i \left[ 1 - y_i (w_0 + w^T x_i) \right] \right\}
\]

\[
L(w, w_0; \alpha)
\]
Review: SVM optimization (separable case)

\[
\begin{align*}
\max \{ \alpha_i \geq 0 \} \min_{(w, w_0)} & \left\{ \frac{1}{2} \|w\|^2 + \sum_{i=1}^{n} \alpha_i \left[ 1 - y_i (w_0 + w^T x_i) \right] \right\} \\
& L(w, w_0; \alpha)
\end{align*}
\]

First, we fix \( \alpha \) and minimize \( L(w, w_0; \alpha) \) w.r.t. \( w, w_0 \):

\[
\frac{\partial}{\partial w} L(w, w_0; \alpha) = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0,
\]

\[
\frac{\partial}{\partial w_0} L(w, w_0; \alpha) = - \sum_{i=1}^{n} \alpha_i y_i = 0.
\]

\[
 w(\alpha) = \sum_{i=1}^{n} \alpha_i y_i x_i, \quad \sum_{i=1}^{n} \alpha_i y_i = 0.
\]
Review: SVM optimization (separable case)

\[ \mathbf{w}(\alpha) = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i, \quad \sum_{i=1}^{n} \alpha_i y_i = 0. \]

Now we can substitute this solution into

\[
\max \left\{ \alpha_i \geq 0, \sum \alpha_i y_i = 0 \right\} \left\{ \frac{1}{2} \| \mathbf{w}(\alpha) \|^2 + \sum_{i=1}^{n} \alpha_i \left[ 1 - y_i (w_0(\alpha) + \mathbf{w}(\alpha)^T \mathbf{x}_i) \right] \right\}
\]
Review: SVM optimization (separable case)

\[ w(\alpha) = \sum_{i=1}^{n} \alpha_i y_i x_i, \quad \sum_{i=1}^{n} \alpha_i y_i = 0. \]

Now we can substitute this solution into

\[
\max_{\{\alpha_i \geq 0, \sum \alpha_i y_i = 0\}} \left\{ \frac{1}{2} \|w(\alpha)\|^2 + \sum_{i=1}^{n} \alpha_i \left[ 1 - y_i (w_0(\alpha) + w(\alpha)^T x_i) \right] \right\}
\]

\[
= \max_{\{\alpha_i \geq 0, \sum \alpha_i y_i = 0\}} \left\{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \right\}.
\]
Dual optimization problem

\[
\begin{align*}
\max & \left\{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \right\} \\
\text{subject to} & \sum_{i=1}^{n} \alpha_i y_i = 0, \; \alpha_i \geq 0 \; \text{for all} \; i = 1, \ldots, n.
\end{align*}
\]

Solving this quadratic program yields the optimal \( \alpha \). We substitute it back to get \( w \):

\[
\mathbf{w} = \mathbf{w}(\alpha) = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i
\]
Review: SVM optimization (separable case)

Dual optimization problem

\[
\max \left\{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \right\}
\]

subject to \( \sum_{i=1}^{n} \alpha_i y_i = 0, \; \alpha_i \geq 0 \) for all \( i = 1, \ldots, n \).

Solving this quadratic program yields the optimal \( \alpha \). We substitute it back to get \( w \):

\[
w = w(\alpha) = \sum_{i=1}^{n} \alpha_i y_i x_i
\]

What is the structure of the solution?
Review: SVM classification

\[ \mathbf{w} = \sum_{\alpha_i > 0} \alpha_i y_i \mathbf{x}_i. \]

Given a test example \( \mathbf{x} \), how is it classified?
Review: SVM classification

\[ \mathbf{w} = \sum_{\alpha_i > 0} \alpha_i y_i \mathbf{x}_i. \]

Given a test example \( \mathbf{x} \), how is it classified?

\[ y = \text{sign} \left( w_0 + \mathbf{w}^T \mathbf{x} \right) \]

\[ = \text{sign} \left( w_0 + \left( \sum_{\alpha_i > 0} \alpha_i y_i \mathbf{x}_i \right)^T \mathbf{x} \right) \]

\[ = \text{sign} \left( w_0 + \sum_{\alpha_i > 0} \alpha_i y_i \mathbf{x}_i^T \mathbf{x} \right) \]

The classifier is based on the expansion in terms of dot products of \( \mathbf{x} \) with support vectors.
What if the training data are not linearly separable?
What if the training data are not linearly separable?

Basic idea: minimize

$$\frac{1}{2} \| w \|^2 + C' \text{(penalty for violating margin constraints)}.$$
Rewrite the constraints with slack variables $\xi_i \geq 0$:

$$\min_{(w,w_0)} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i$$

subject to $y_i (w_0 + w^T x_i) - 1 + \xi_i \geq 0$. Whenever margin is $\geq 1$ (original constraint is satisfied), $\xi_i = 0$. Whenever margin is $< 1$ (constraint violated), pay linear penalty: $\xi_i = 1 - y_i (w_0 + w^T x_i)$.Penalty function $\max(0, 1 - y_i (w_0 + w^T x_i))$.
Non-separable case

Rewrite the constraints with slack variables $\xi_i \geq 0$:

$$
\min_{(\mathbf{w},w_0)} \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{n} \xi_i
$$

subject to $y_i \left( w_0 + \mathbf{w}^T \mathbf{x}_i \right) - 1 + \xi_i \geq 0$.

- Whenever margin is $\geq 1$ (original constraint is satisfied), $\xi_i = 0$. 
Non-separable case

Rewrite the constraints with slack variables $\xi_i \geq 0$:

$$\min_{(w, w_0)} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i$$

subject to $y_i (w_0 + w^T x_i) - 1 + \xi_i \geq 0$.

- Whenever margin is $\geq 1$ (original constraint is satisfied), $\xi_i = 0$.
- Whenever margin is $< 1$ (constraint violated), pay linear penalty: $\xi_i = 1 - y_i (w_0 + w^T x_i)$.
Non-separable case

Rewrite the constraints with slack variables $\xi_i \geq 0$:

$$\min_{(w,w_0)} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i$$

subject to $y_i (w_0 + w^T x_i) - 1 + \xi_i \geq 0$.

- Whenever margin is $\geq 1$ (original constraint is satisfied), $\xi_i = 0$.
- Whenever margin is $< 1$ (constraint violated), pay linear penalty: $\xi_i = 1 - y_i (w_0 + w^T x_i)$.

**Penalty function**

$$\max(0, 1 - y_i (w_0 + w^T x_i))$$
Review: Hinge loss

\[ \max \left( 0, 1 - y_i (w_0 + w^T x_i) \right) \]
Connection between SVMs and logistic regression

Support vector machines:

Hinge loss: \( \max \left( 0, 1 - y_i (w_0 + w^T x_i) \right) \)

Logistic regression:

\[
P(y_i | x_i; w, w_0) = \frac{1}{1 + e^{-y_i (w_0 + w^T x_i)}}
\]

Log loss: \( \log \left( 1 + e^{-y_i (w_0 + w^T x_i)} \right) \)
Dual problem:

$$\begin{align*}
\max \left\{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \right\}
\end{align*}$$

subject to $\sum_{i=1}^{n} \alpha_i y_i = 0$, $0 \leq \alpha_i \leq C$, for all $i = 1, \ldots, N$.

- $\alpha_i = 0$: not support vector.
- $0 < \alpha_i < C$: SV on the margin, $\xi_i = 0$.
- $\alpha_i = C$: over the margin, either misclassified ($\xi_i > 1$) or not ($0 < \xi_i \leq 1$).
General idea: try to map the original input space into a high-dimensional feature space where the data is separable.

\[ \Phi: \mathbf{x} \rightarrow \varphi(\mathbf{x}) \]
Example of nonlinear mapping

Not separable in 1D:

\[ \phi(x) = (x, x^2). \]
Example of nonlinear mapping

Not separable in 1D:

Separable in 2D:

What is $\phi(x)$?

$\phi(x) = (x, x^2)$. 
Example of nonlinear mapping

Not separable in 1D:

Separable in 2D:

What is $\phi(x)$?

$\phi(x) = (x, x^2)$. 
Example of nonlinear mapping

Not separable in 1D:

Separable in 2D:

What is $\phi(x)$? $\phi(x) = (x, x^2)$. 
Example of nonlinear mapping

Consider the mapping:

$$\phi : [x_1, x_2]^T \rightarrow [1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2]^T.$$  

The (linear) SVM classifier in the feature space:

$$\hat{y} = \text{sign} \left( \hat{w}_0 + \sum_{\alpha_i > 0} \alpha_i y_i \phi(x_i)^T \phi(x) \right)$$

The dot product in the feature space:

$$\phi(x)^T \phi(z) = 1 + 2x_1z_1 + 2x_2z_2 + x_1^2z_1^2 + x_2^2z_2^2 + 2x_1x_2z_1z_2 = (1 + x^T z)^2.$$
We defined a non-linear mapping into feature space

\[ \phi : [x_1, x_2]^T \rightarrow [1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2]^T \]

and saw that \( \phi(x)^T\phi(z) = K(x, z) \) using the kernel

\[ K(x, z) = (1 + x^Tz)^2. \]

I.e., we can calculate dot products in the feature space implicitly, without ever writing the feature expansion!
The kernel trick

- Replace dot products in the SVM formulation with kernel values.

- The optimization problem:

\[
\max \left\{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \right\}
\]

- Need to compute pairwise kernel values for training data.
The kernel trick

- Replace dot products in the SVM formulation with kernel values.

- The optimization problem:

\[
\max \left\{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \right\}
\]

- Need to compute pairwise kernel values for training data.

- The classifier now defines a **nonlinear** decision boundary in the original space:

\[
\hat{y} = \text{sign} \left( \hat{w}_0 + \sum_{\alpha_i > 0} \alpha_i y_i K(x_i, x) \right)
\]

- Need to compute \( K(x_i, x) \) for all SVs \( x_i \).
What kind of function $K$ is a valid kernel, i.e. such that there exists a feature space $\Phi(x)$ in which $K(x, z) = \phi(x)^T \phi(z)$?

**Theorem due to Mercer (1930s)**

$K$ must be

- continuous;
- symmetric: $K(x, z) = K(z, x)$;
- positive definite: for any $x_1, \ldots, x_N$, the kernel matrix

$$K = \begin{bmatrix}
K(x_1, x_1) & K(x_1, x_2) & K(x_1, x_N) \\
\vdots & \ddots & \vdots \\
K(x_N, x_1) & K(x_N, x_2) & K(x_N, x_N)
\end{bmatrix}$$

must be positive definite.
The linear kernel:

\[ K(x, z) = x^T z. \]

This leads to the original, linear SVM.
Some popular kernels

- The **linear kernel**:
  \[ K(x, z) = x^T z. \]
  This leads to the original, linear SVM.

- The **polynomial kernel**:
  \[ K(x, z; c, d) = (c + x^T z)^d. \]
  We can write the expansion explicitly, by concatenating powers up to \( d \) and multiplying by appropriate weights.
Example: SVM with polynomial kernel

Source: G. Shakhnarovich

- **linear**
- $2^{nd}$ order polynomial
- $4^{th}$ order polynomial
- $8^{th}$ order polynomial
Radial basis function kernel

\[ K(x, z; \sigma) = \exp \left( -\frac{1}{\sigma^2} \| x - z \|^2 \right). \]

- The RBF kernel is a measure of similarity between two examples.
  - The mapping \( \phi(x) \) is infinite-dimensional!

- What is the role of parameter \( \sigma \)?
Radial basis function kernel

\[ K(x, z; \sigma) = \exp \left( -\frac{1}{\sigma^2} \| x - z \|^2 \right). \]

- The RBF kernel is a measure of similarity between two examples.
  - The mapping \( \phi(x) \) is infinite-dimensional!

- What is the role of parameter \( \sigma \)?
  - Consider \( \sigma \to 0 \).
Radial basis function kernel

The RBF kernel is a measure of similarity between two examples.

- The mapping \( \phi(x) \) is infinite-dimensional!

What is the role of parameter \( \sigma \)?

- Consider \( \sigma \to 0 \). Then \( K(x_i, x; \sigma) \to 1 \) if \( x = z \) or \( 0 \) if \( x \neq z \).
Radial basis function kernel

\[ K(x, z; \sigma) = \exp\left(-\frac{1}{\sigma^2} \|x - z\|^2\right). \]

- The RBF kernel is a measure of similarity between two examples.
  - The mapping \( \phi(x) \) is infinite-dimensional!

- What is the role of parameter \( \sigma \)?
  - Consider \( \sigma \to 0 \). Then \( K(x_i, x; \sigma) \to 1 \) if \( x = z \) or \( 0 \) if \( x \neq z \). The SVM simply “memorizes” the training data (overfitting, lack of generalization).
Radial basis function kernel

\[ K(x, z; \sigma) = \exp \left( -\frac{1}{\sigma^2} \|x - z\|^2 \right) . \]

- The RBF kernel is a measure of similarity between two examples.
  - The mapping \( \phi(x) \) is infinite-dimensional!

What is the role of parameter \( \sigma \)?

- Consider \( \sigma \to 0 \). Then \( K(x_i, x; \sigma) \to 1 \) if \( x = z \) or \( 0 \) if \( x \neq z \). The SVM simply “memorizes” the training data (overfitting, lack of generalization).
- What about \( \sigma \to \infty \)?
Radial basis function kernel

\[ K(x, z; \sigma) = \exp \left( -\frac{1}{\sigma^2} \|x - z\|^2 \right). \]

- The RBF kernel is a measure of similarity between two examples.
  - The mapping \( \phi(x) \) is infinite-dimensional!

- What is the role of parameter \( \sigma \)?
  - Consider \( \sigma \to 0 \). Then \( K(x_i, x; \sigma) \to 1 \) if \( x = z \) or \( 0 \) if \( x \neq z \).
    The SVM simply “memorizes” the training data (overfitting, lack of generalization).
  - What about \( \sigma \to \infty \)? Then \( K(x, z) \to 1 \) for all \( x, z \).
Radial basis function kernel

\[ K(x, z; \sigma) = \exp\left(-\frac{1}{\sigma^2} \|x - z\|^2\right). \]

- The RBF kernel is a measure of similarity between two examples.
  - The mapping \( \phi(x) \) is infinite-dimensional!

- What is the role of parameter \( \sigma \)?
  - Consider \( \sigma \to 0 \). Then \( K(x_i, x; \sigma) \to 1 \) if \( x = z \) or \( 0 \) if \( x \neq z \).
    The SVM simply “memorizes” the training data (overfitting, lack of generalization).
  - What about \( \sigma \to \infty \)? Then \( K(x, z) \to 1 \) for all \( x, z \). The SVM underfits.
SVM with RBF (Gaussian) kernels

Note: some SV here not close to the boundary
Let $K_1$ and $K_2$ be Mercer kernels. Then the following functions are also Mercer kernels:

- $K(x, z) = K_1(x, z) + K_2(x, z)$
- $K(x, z) = aK_1(x, z)$ ($a$ is a positive scalar)
- $K(x, z) = K_1(x, z) K_2(x, z)$
- $K(x, z) = x^T B z$ ($B$ is a symmetric positive semi-definite matrix)
Let $K_1$ and $K_2$ be Mercer kernels. Then the following functions are also Mercer kernels:

- $K(x, z) = K_1(x, z) + K_2(x, z)$
- $K(x, z) = aK_1(x, z)$ ($a$ is a positive scalar)
- $K(x, z) = K_1(x, z) K_2(x, z)$
- $K(x, z) = x^T B z$ ($B$ is a symmetric positive semi-definite matrix)

**Multiple kernel learning**: learn kernel combinations as part of SVM optimization.
Various “direct” formulations exist, but they are not widely used in practice. It is more common to obtain multi-class classifiers by combining two-class SVMs in various ways.
Multi-class SVMs

- Various “direct” formulations exist, but they are not widely used in practice. It is more common to obtain multi-class classifiers by combining two-class SVMs in various ways.

- **One vs. others:**
  - **Training:** learn an SVM for each class vs. the others
  - **Testing:** apply each SVM to test example and assign to it the class of the SVM that returns the highest decision value

- Error-correcting codes, decision trees/DAGs..
Multi-class SVMs

- Various “direct” formulations exist, but they are not widely used in practice. It is more common to obtain multi-class classifiers by combining two-class SVMs in various ways.

  - **One vs. others:**
    - **Training:** learn an SVM for each class vs. the others
    - **Testing:** apply each SVM to test example and assign to it the class of the SVM that returns the highest decision value

  - **One vs. one:**
    - **Training:** learn an SVM for each pair of classes
    - **Testing:** each learned SVM “votes” for a class to assign to the test example
Multi-class SVMs

- Various “direct” formulations exist, but they are not widely used in practice. It is more common to obtain multi-class classifiers by combining two-class SVMs in various ways.

**One vs. others:**
- **Training:** learn an SVM for each class vs. the others
- **Testing:** apply each SVM to test example and assign to it the class of the SVM that returns the highest decision value

**One vs. one:**
- **Training:** learn an SVM for each pair of classes
- **Testing:** each learned SVM “votes” for a class to assign to the test example

- Error-correcting codes, decision trees/DAGs..
The model: \( f(x) = w_0 + w^T x \)
The model: \( f(x) = w_0 + \mathbf{w}^T \mathbf{x} \)

Instead of the margin around the predicted decision boundary, we have \( \epsilon \)-tube around the predicted function.
Optimization: introduce constraints and slack variables for going above or below the tube.

\[
\min_{w_0, w} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} (\xi_i + \hat{\xi}_i)
\]

subject to \((w_0 + w^T x_i) - y_i \leq \epsilon + \xi_i\),
\[y_i - (w_0 + w^T x_i) \leq \epsilon + \hat{\xi}_i,\]
\[\xi_i, \hat{\xi}_i \geq 0, i = 1, \ldots, n.\]
Support Vector Regression: Dual problem

\[
\max_{\alpha} \sum_{i=1}^{n} (\hat{\alpha}_i - \alpha_i) y_i - \epsilon \sum_{i=1}^{n} (\hat{\alpha}_i + \alpha_i) - \frac{1}{2} \sum_{i,j=1}^{n} (\hat{\alpha}_i - \alpha_i)(\hat{\alpha}_j - \alpha_j) x_i^T x_j ,
\]

subject to \( 0 \leq \hat{\alpha}_i, \alpha_i \leq C, \sum_{i=1}^{n} (\hat{\alpha}_i - \alpha_i) = 0, i = 1, \ldots, n. \)

Note that at the solution, we must have \( \xi_i \hat{\xi}_i = 0, \alpha_i \hat{\alpha}_i = 0. \)
Support Vector Regression: Dual problem

\[
\max_{\alpha} \sum_{i=1}^{n} (\hat{\alpha}_i - \alpha_i) y_i - \epsilon \sum_{i=1}^{n} (\hat{\alpha}_i + \alpha_i) - \frac{1}{2} \sum_{i,j=1}^{n} (\hat{\alpha}_i - \alpha_i)(\hat{\alpha}_j - \alpha_j)x_i^T x_j,
\]

subject to \(0 \leq \hat{\alpha}_i, \alpha_i \leq C, \sum_{i=1}^{n} (\hat{\alpha}_i - \alpha_i) = 0, i = 1, \ldots, n.\)

Note that at the solution, we must have \(\xi_i \hat{\xi}_i = 0, \alpha_i \hat{\alpha}_i = 0.\)

We can let \(\beta_i = \hat{\alpha}_i - \alpha_i\) and simplify:

\[
\max_{\beta} \sum_{i=1}^{n} y_i \beta_i - \epsilon \sum_{i=1}^{n} |\beta_i| - \frac{1}{2} \sum_{i,j=1}^{n} \beta_i \beta_j x_i^T x_j,
\]

subject to \(-C \leq \beta_i \leq C, \sum_{i=1}^{n} \beta_i = 0, i = 1, \ldots, n.\)
Support Vector Regression: Dual problem

\[
\begin{align*}
\max_{\alpha} & \sum_{i=1}^{n} (\hat{\alpha}_i - \alpha_i) y_i - \epsilon \sum_{i=1}^{n} (\hat{\alpha}_i + \alpha_i) - \frac{1}{2} \sum_{i,j=1}^{n} (\hat{\alpha}_i - \alpha_i)(\hat{\alpha}_j - \alpha_j) x_i^T x_j, \\
\text{subject to} & \quad 0 \leq \hat{\alpha}_i, \alpha_i \leq C, \sum_{i=1}^{n} (\hat{\alpha}_i - \alpha_i) = 0, i = 1, \ldots, n.
\end{align*}
\]

Note that at the solution, we must have \(\xi_i \hat{\xi}_i = 0\), \(\alpha_i \hat{\alpha}_i = 0\).

We can let \(\beta_i = \hat{\alpha}_i - \alpha_i\) and simplify:

\[
\begin{align*}
\max_{\beta} & \sum_{i=1}^{n} y_i \beta_i - \epsilon \sum_{i=1}^{n} |\beta_i| - \frac{1}{2} \sum_{i,j=1}^{n} \beta_i \beta_j x_i^T x_j, \\
\text{subject to} & \quad -C \leq \beta_i \leq C, \sum_{i=1}^{n} \beta_i = 0, i = 1, \ldots, n.
\end{align*}
\]

Then \(f(x) = w_0^* + \sum_{i=1}^{n} \beta_i^* x_i^T x_j\), where \(w_0^*\) is chosen so that \(f(x_i) - y_i = -\epsilon\) for any \(i\) with \(0 < \beta_i^* < C\).
Main ideas:
Main ideas:
  - Large margin classification
Main ideas:
- Large margin classification
- The kernel trick
Main ideas:

- Large margin classification
- The kernel trick

What does the complexity/generalization ability of SVMs depend on?
Main ideas:
  - Large margin classification
  - The kernel trick
What does the complexity/generalization ability of SVMs depend on?
  - The constant $C$
Main ideas:
- Large margin classification
- The kernel trick

What does the complexity/generalization ability of SVMs depend on?
- The constant $C$
- Choice of kernel and kernel parameters
Main ideas:

- Large margin classification
- The kernel trick

What does the complexity/generalization ability of SVMs depend on?

- The constant $C$
- Choice of kernel and kernel parameters
- Number of support vectors
Main ideas:
- Large margin classification
- The kernel trick

What does the complexity/generalization ability of SVMs depend on?
- The constant $C$
- Choice of kernel and kernel parameters
- Number of support vectors

A crucial component: good QP solver.

Tons of off-the-shelf packages.
Advantages and disadvantages of SVMs

Advantages:

- One of the most successful ML techniques!
- Good generalization ability for small training sets
- Kernel trick is powerful and flexible
- Margin-based formalism can be extended to a large class of problems (regression, structured prediction, etc.)

Disadvantages:

- Computational and storage complexity of training: quadratic in the size of the training set
- In the worst case, can degenerate to nearest neighbor (every training point a support vector), but is much slower to train
- No direct multi-class formulation
Advantages and disadvantages of SVMs

Advantages:

- One of the most successful ML techniques!
- Good generalization ability for small training sets
- Kernel trick is powerful and flexible
- Margin-based formalism can be extended to a large class of problems (regression, structured prediction, etc.)

Disadvantages:

- Computational and storage complexity of training:

  \( \text{quadratic} \)
Advantages and disadvantages of SVMs

Advantages:

- One of the most successful ML techniques!
- Good generalization ability for small training sets
- Kernel trick is powerful and flexible
- Margin-based formalism can be extended to a large class of problems (regression, structured prediction, etc.)

Disadvantages:

- Computational and storage complexity of training: quadratic in the size of the training set
Advantages and disadvantages of SVMs

Advantages:
- One of the most successful ML techniques!
- Good generalization ability for small training sets
- Kernel trick is powerful and flexible
- Margin-based formalism can be extended to a large class of problems (regression, structured prediction, etc.)

Disadvantages:
- Computational and storage complexity of training: quadratic in the size of the training set
- In the worst case, can degenerate to nearest neighbor (every training point a support vector), but is much slower to train
Advantages and disadvantages of SVMs

Advantages:
- One of the most successful ML techniques!
- Good generalization ability for small training sets
- Kernel trick is powerful and flexible
- Margin-based formalism can be extended to a large class of problems (regression, structured prediction, etc.)

Disadvantages:
- Computational and storage complexity of training: quadratic in the size of the training set
- In the worst case, can degenerate to nearest neighbor (every training point a support vector), but is much slower to train
- No direct multi-class formulation