Machine Learning

Support Vector Machines

FAST DISCOVERING THE FUTURE

Topics of previous lectures

- ✓ Ingredients of Machine Learning
- ✓ Classification Basics
- ✓ Basic Linear Classifier
- ✓ K-Nearest Neighbours Classifier
- ✓ Naive Bayes Classifier
- ✓ Linear and Quadratic Discriminant Analysis

Topics of today's lecture

- Convex Optimization
- Support Vector Machine (SVM)
- Hard-margin SVM
- Soft-margin SVM

Background for Constrained Optimization

Consider the following optimization problem

$$\min_{x,y} f(x, y)$$

$$\text{subject to } g(x, y) = c$$

Background for Constrained Optimization

Consider the following optimization problem

$$\min_{x,y} f(x, y)$$

$$\text{subject to } g(x, y) = c$$

We can solve this task with the method of **Lagrange multipliers**:

Background for Constrained Optimization

Consider the following optimization problem

$$\min_{x,y} f(x, y)$$

$$\text{subject to } g(x, y) = c$$

We can solve this task with the method of **Lagrange multipliers**:

- form the Lagrange function (Lagragian):

$$\Lambda(x, y, \lambda) = f(x, y) - \lambda(g(x, y) - c)$$

  where $\lambda \in \mathbb{R}$ is the Lagrange multiplier

Background for Constrained Optimization

Consider the following optimization problem

$$\min_{x,y} f(x,y)$$

$$\text{subject to } g(x,y) = c$$

We can solve this task with the method of **Lagrange multipliers**:

- form the Lagrange function (Lagragian):

$$\Lambda(x,y,\lambda) = f(x,y) - \lambda(g(x,y) - c)$$

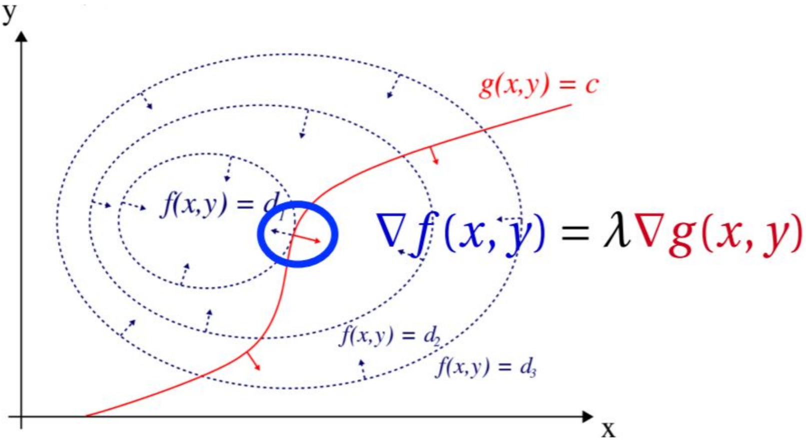
  where $\lambda \in \mathbb{R}$ is the Lagrange multiplier

- Solve the unconstrained problem: $\nabla\Lambda(x,y,\lambda) = 0$, which is equivalent to

$$\nabla_{x,y}\Lambda(x,y,\lambda) = \nabla f(x,y) - \lambda\nabla g(x,y) = 0$$

$$\nabla_{\lambda}\Lambda(x,y,\lambda) = g(x,y) - c = 0$$

Background for Constrained Optimization



Background for Constrained Optimization

The following optimization problem

$$\min_{\mathbf{x}} f(\mathbf{x})$$

$$\text{subject to } g_i(\mathbf{x}) \leq 0, \quad \forall i = 1, \ldots, m$$

is known as the **primal problem**, corresponding to *primal variables* $\mathbf{x}$.

Background for Constrained Optimization

The following optimization problem

$$\min_{\mathbf{x}} f(\mathbf{x})$$

$$\text{subject to } g_i(\mathbf{x}) \leq 0, \quad \forall i = 1, \ldots, m$$

is known as the **primal problem**, corresponding to *primal variables* $\mathbf{x}$. The associated Lagragian **dual problem** is given by

$$\max_{\boldsymbol{\lambda} \in \mathbb{R}^m} \min_{\mathbf{x} \in \mathbb{R}^d} \Lambda(\mathbf{x}, \boldsymbol{\lambda})$$

$$\text{subject to } \boldsymbol{\lambda} \geq 0$$

where $\boldsymbol{\lambda}$ are the *dual variables*.

Background for Convex Optimization

- When $f(\cdot)$ is a convex function and the constraints form a convex set in the primal problem, then we will call it a **convex optimization** problem.

Background for Convex Optimization

- When $f(\cdot)$ is a convex function and the constraints form a convex set in the primal problem, then we will call it a **convex optimization** problem.
- In this case, the optimal solution of the dual problem is the same as the optimal solution of the primal problem (Karush–Kuhn–Tucker (KKT) theorem).

Background for Convex Optimization

- When $f(\cdot)$ is a convex function and the constraints form a convex set in the primal problem, then we will call it a **convex optimization** problem.
- In this case, the optimal solution of the dual problem is the same as the optimal solution of the primal problem (Karush–Kuhn–Tucker (KKT) theorem).

Definition (Convex set)

A set C is a convex set if $\forall x, y \in C$ and $\forall \theta \in [0, 1]$, we have

$$\theta x + (1 - \theta) y \in C$$

Background for Convex Optimization

- When $f(\cdot)$ is a convex function and the constraints form a convex set in the primal problem, then we will call it a **convex optimization** problem.
- In this case, the optimal solution of the dual problem is the same as the optimal solution of the primal problem (Karush–Kuhn–Tucker (KKT) theorem).

Definition (Convex set)

A set C is a convex set if $\forall x, y \in C$ and $\forall \theta \in [0, 1]$, we have

$$\theta x + (1 - \theta) y \in C$$

Definition (Convex function)

Let $f : X \to \mathbb{R}$ be a function such that X is a convex set, then f is a convex (concave) function if $\forall \mathbf{x}_1, \mathbf{x}_2 \in X$ and $\forall \theta \in [0, 1]$, we have

$$f(\theta \mathbf{x}_1 + (1 - \theta)\mathbf{x}_2) \leq \theta f(\mathbf{x}_1) + (1 - \theta) f(\mathbf{x}_2)$$

Background for Convex Optimization

Theorem 1

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is differentiable, then $f(\mathbf{x})$ is convex iff $\forall \mathbf{x}_1, \mathbf{x}_2$

$$f(\mathbf{x}_2) \geq f(\mathbf{x}_1) + \nabla_{\mathbf{x}} f(\mathbf{x}_1)^T (\mathbf{x}_2 - \mathbf{x}_1).$$

Background for Convex Optimization

Theorem 1

Let $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable, then $f(\mathbf{x})$ is convex iff $\forall \mathbf{x}_1, \mathbf{x}_2$

$$f(\mathbf{x}_2) \geq f(\mathbf{x}_1) + \nabla_{\mathbf{x}} f(\mathbf{x}_1)^T (\mathbf{x}_2 - \mathbf{x}_1).$$

Theorem 2

Let $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable, then $f(\mathbf{x})$ is convex iff the Hesian matrix $\nabla^2_{\mathbf{x}} f(\mathbf{x})$ is positive semidefinite.

Background for Convex Optimization

Theorem 1

Let $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable, then $f(\mathbf{x})$ is convex iff $\forall \mathbf{x}_1, \mathbf{x}_2$

$$f(\mathbf{x}_2) \geq f(\mathbf{x}_1) + \nabla_{\mathbf{x}} f(\mathbf{x}_1)^T (\mathbf{x}_2 - \mathbf{x}_1).$$

Theorem 2

Let $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable, then $f(\mathbf{x})$ is convex iff the Hesian matrix $\nabla^2_{\mathbf{x}} f(\mathbf{x})$ is positive semidefinite.

Now let's look at two well-known classes of convex optimization problems.

Linear Programming

Consider the case when the objective function is linear

$$\min_{\mathbf{x}\in\mathbb{R}^d} \mathbf{c}^T\mathbf{x}$$

$$\text{subject to } \boldsymbol{A}\mathbf{x} \leq \mathbf{b},$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times d}, \quad \mathbf{c} \in \mathbb{R}^d$ and $\mathbf{b} \in \mathbb{R}^m$

Consider the case when the objective function is linear

$$\min_{\mathbf{x}\in\mathbb{R}^d} \mathbf{c}^T\mathbf{x}$$

$$\text{subject to } \boldsymbol{A}\mathbf{x} \leq \mathbf{b},$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times d}, \quad \mathbf{c} \in \mathbb{R}^d$ and $\mathbf{b} \in \mathbb{R}^m$
The Lagragian will be

$$\Lambda(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{c}^T\mathbf{x} + \boldsymbol{\lambda}^T(\boldsymbol{A}\mathbf{x} - \mathbf{b}) = (\mathbf{c} + \boldsymbol{A}^T\boldsymbol{\lambda})^T\mathbf{x} - \boldsymbol{\lambda}^T\mathbf{b},$$

where $\boldsymbol{\lambda} \in \mathbb{R}^m$ is the vector of non-negative Lagrange multipliers.

Linear Programming

Consider the case when the objective function is linear

$$\min_{\mathbf{x}\in\mathbb{R}^d} \mathbf{c}^T\mathbf{x}$$

$$\text{subject to } \boldsymbol{A}\mathbf{x} \leq \mathbf{b},$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times d}, \quad \mathbf{c} \in \mathbb{R}^d$ and $\mathbf{b} \in \mathbb{R}^m$
The Lagragian will be

$$\Lambda(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{c}^T\mathbf{x} + \boldsymbol{\lambda}^T(\boldsymbol{A}\mathbf{x} - \mathbf{b}) = (\mathbf{c} + \boldsymbol{A}^T\boldsymbol{\lambda})^T\mathbf{x} - \boldsymbol{\lambda}^T\mathbf{b},$$

where $\boldsymbol{\lambda} \in \mathbb{R}^m$ is the vector of non-negative Lagrange multipliers.

$$\nabla_{\mathbf{x}}\Lambda(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{c} + \boldsymbol{A}^T\boldsymbol{\lambda} = 0$$

Linear Programming

Consider the case when the objective function is linear

$$\min_{\mathbf{x}\in\mathbb{R}^d} \mathbf{c}^T\mathbf{x}$$

$$\text{subject to } \boldsymbol{A}\mathbf{x} \leq \mathbf{b},$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times d}, \quad \mathbf{c} \in \mathbb{R}^d$ and $\mathbf{b} \in \mathbb{R}^m$

The Lagragian will be

$$\Lambda(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{c}^T\mathbf{x} + \boldsymbol{\lambda}^T(\boldsymbol{A}\mathbf{x} - \mathbf{b}) = (\mathbf{c} + \boldsymbol{A}^T\boldsymbol{\lambda})^T\mathbf{x} - \boldsymbol{\lambda}^T\mathbf{b},$$

where $\boldsymbol{\lambda} \in \mathbb{R}^m$ is the vector of non-negative Lagrange multipliers.

$$\nabla_{\mathbf{x}}\Lambda(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{c} + \boldsymbol{A}^T\boldsymbol{\lambda} = 0$$

The dual Lagragian problem will be

$$\max_{\lambda\in\mathbb{R}^m} -\boldsymbol{\lambda}^T\mathbf{b}$$

$$\text{subject to } \mathbf{c} + A^T\boldsymbol{\lambda} = 0, \quad \boldsymbol{\lambda} \geq 0$$

Quadratic Programming

Consider the case when the objective function is quadratic

$$\min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x}$$

$$\text{subject to } \boldsymbol{A}\mathbf{x} \leq \mathbf{b},$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times d}, \quad \mathbf{c} \in \mathbb{R}^d, \quad \mathbf{b} \in \mathbb{R}^m$ and $\mathbf{Q} \in \mathbb{R}^{d\times d}$ is positive definite.

Quadratic Programming

Consider the case when the objective function is quadratic

$$\min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x}$$

$$\text{subject to } \boldsymbol{A}\mathbf{x} \leq \mathbf{b},$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times d}, \quad \mathbf{c} \in \mathbb{R}^d, \quad \mathbf{b} \in \mathbb{R}^m$ and $\mathbf{Q} \in \mathbb{R}^{d\times d}$ is positive definite. The Lagrangian will be

$$\Lambda(\mathbf{x}, \boldsymbol{\lambda}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + c^T\mathbf{x} + \boldsymbol{\lambda}^T(\boldsymbol{A}\mathbf{x} - \mathbf{b}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + (c + \boldsymbol{A}^T\boldsymbol{\lambda})^T\mathbf{x} - \boldsymbol{\lambda}^T\mathbf{b},$$

Quadratic Programming

Consider the case when the objective function is quadratic

$$\min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x}$$

$$\text{subject to } \boldsymbol{A}\mathbf{x} \leq \mathbf{b},$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times d}, \quad \mathbf{c} \in \mathbb{R}^d, \quad \mathbf{b} \in \mathbb{R}^m$ and $\mathbf{Q} \in \mathbb{R}^{d\times d}$ is positive definite. The Lagragian will be

$$\Lambda(\mathbf{x},\boldsymbol{\lambda}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + c^T\mathbf{x} + \boldsymbol{\lambda}^T(\boldsymbol{A}\mathbf{x} - \mathbf{b}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + (c + \boldsymbol{A}^T\boldsymbol{\lambda})^T\mathbf{x} - \boldsymbol{\lambda}^T\mathbf{b},$$

$$\nabla_{\mathbf{x}}\Lambda(\mathbf{x},\boldsymbol{\lambda}) = \mathbf{Q}\mathbf{x} + (\mathbf{c} + \boldsymbol{A}^T\boldsymbol{\lambda}) = 0$$
$$\mathbf{x} = -\mathbf{Q}^{-1}(\mathbf{c} + \boldsymbol{A}^T\boldsymbol{\lambda})$$

Quadratic Programming

Consider the case when the objective function is quadratic

$$\min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x}$$

subject to $\boldsymbol{A}\mathbf{x} \leq \mathbf{b}$,

where $\boldsymbol{A} \in \mathbb{R}^{m\times d}, \quad \mathbf{c} \in \mathbb{R}^d, \quad \mathbf{b} \in \mathbb{R}^m$ and $\mathbf{Q} \in \mathbb{R}^{d\times d}$ is positive definite.
The Lagragian will be

$$\Lambda(\mathbf{x}, \boldsymbol{\lambda}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + c^T\mathbf{x} + \boldsymbol{\lambda}^T(\boldsymbol{A}\mathbf{x} - \mathbf{b}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + (c + \boldsymbol{A}^T\boldsymbol{\lambda})^T\mathbf{x} - \boldsymbol{\lambda}^T\mathbf{b},$$

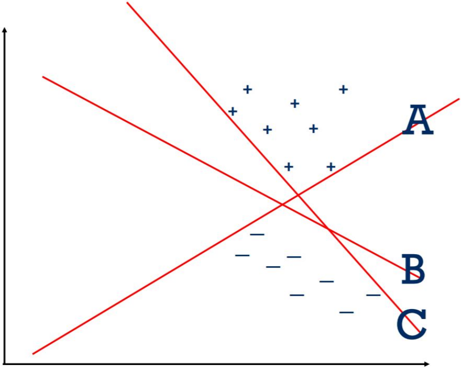
$$\nabla_{\mathbf{x}}\Lambda(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{Q}\mathbf{x} + (\mathbf{c} + \boldsymbol{A}^T\boldsymbol{\lambda}) = 0$$
$$\mathbf{x} = -\mathbf{Q}^{-1}(\mathbf{c} + \boldsymbol{A}^T\boldsymbol{\lambda})$$

The dual Lagragian problem will be

$$\max_{\lambda\in\mathbb{R}^m} -\frac{1}{2}(\mathbf{c} + \boldsymbol{A}^T\boldsymbol{\lambda})^T\mathbf{Q}^{-1}(\mathbf{c} + \boldsymbol{A}^T\boldsymbol{\lambda}) - \boldsymbol{\lambda}^T\mathbf{b}$$

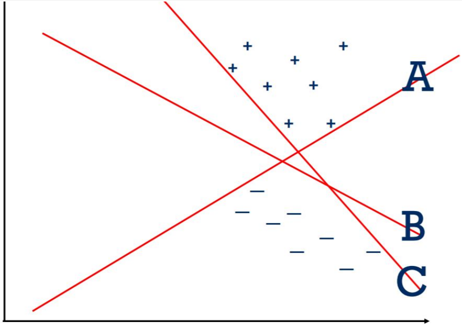
subject to $\boldsymbol{\lambda} \geq 0$

Motivation for SVM



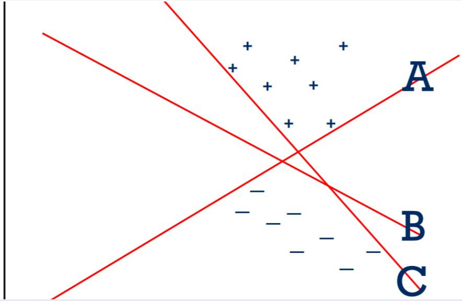
Motivation for SVM

There are infinetely many separating classifiers (decision boundaries) between linearly separable classes. Which one is the best in your opinion?



Motivation for SVM

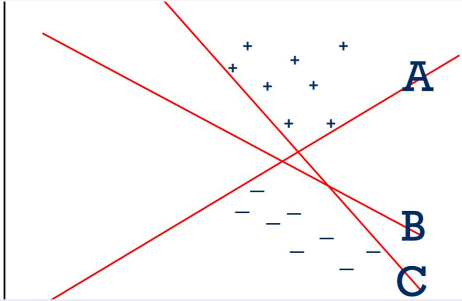
There are infinetely many separating classifiers (decision boundaries) between linearly separable classes. Which one is the best in your opinion?



On training data each is as good as any other

Motivation for SVM

There are infinetely many separating classifiers (decision boundaries) between linearly separable classes. Which one is the best in your opinion?



On training data each is as good as any other

Support Vector Machine learns the separating line B.

Support Vector Machine (SVM)

- Again a linear classification method

Support Vector Machine (SVM)

- Again a linear classification method
- If the classes are linearly separable then SVM finds a separating model

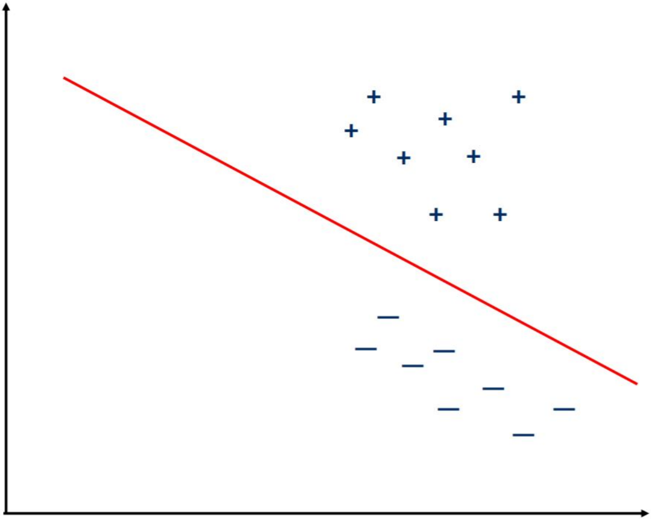
Support Vector Machine (SVM)

- Again a linear classification method
- If the classes are linearly separable then SVM finds a separating model
- In contrast to the perceptron, SVM chooses a particular one among the many

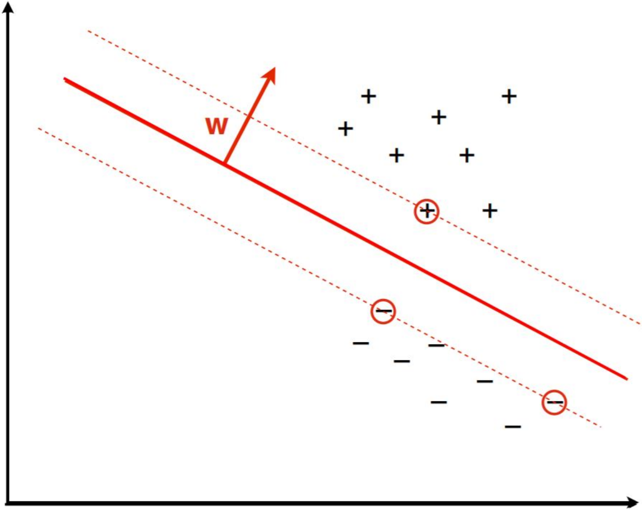
Support Vector Machine (SVM)

- Again a linear classification method
- If the classes are linearly separable then SVM finds a separating model
- In contrast to the perceptron, SVM chooses a particular one among the many
- SVM chooses the linear separating model which has the highest **margin** - distance between the decision boundary and the closest instance

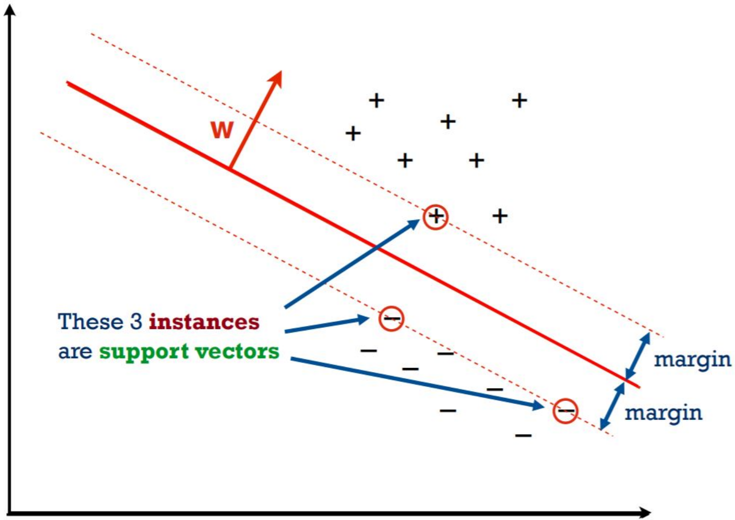
Support Vector Machine (SVM)



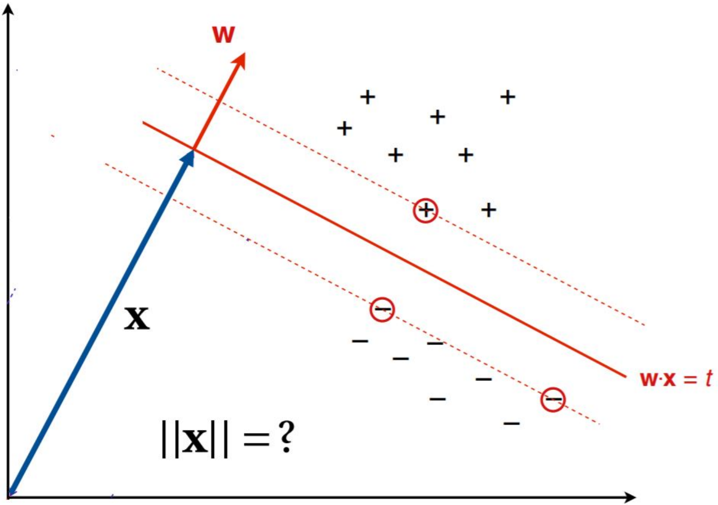
Support Vector Machine (SVM)



Support Vector Machine (SVM)

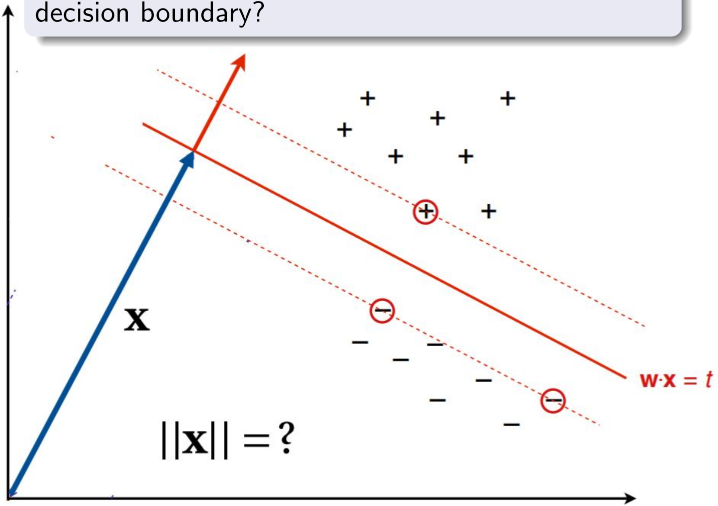


Calculating the margin



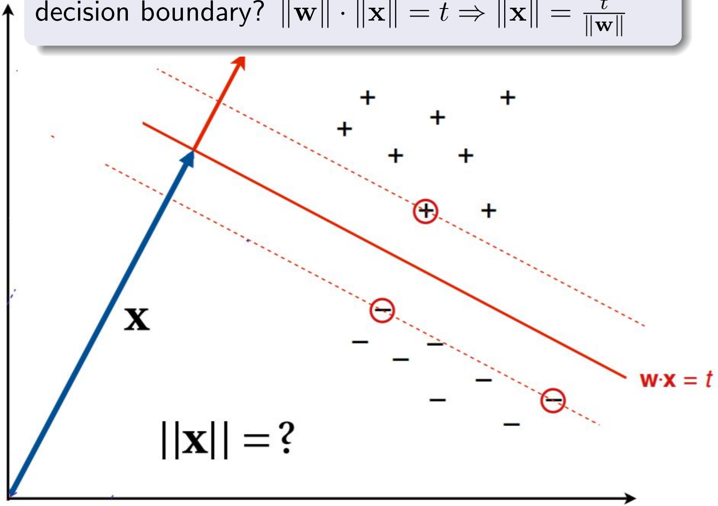
Calculating the margin

What is the distance between origin and the decision boundary?

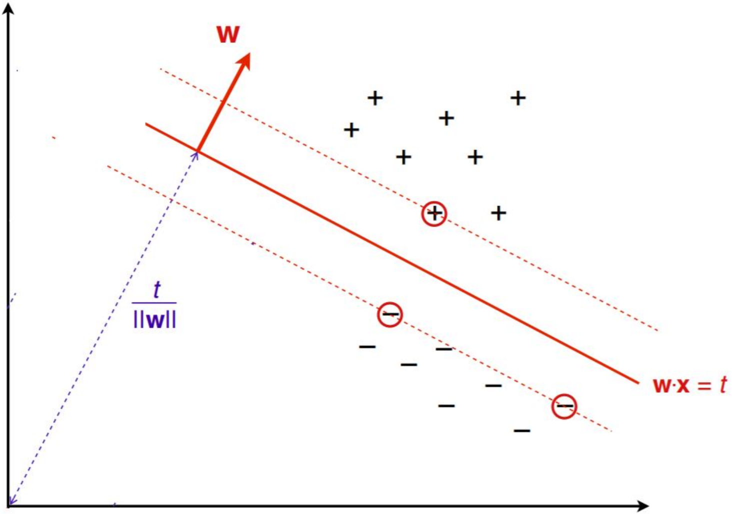


Calculating the margin

What is the distance between origin and the decision boundary? $\|\mathbf{w}\| \cdot \|\mathbf{x}\| = t \Rightarrow \|\mathbf{x}\| = \frac{t}{\|\mathbf{w}\|}$

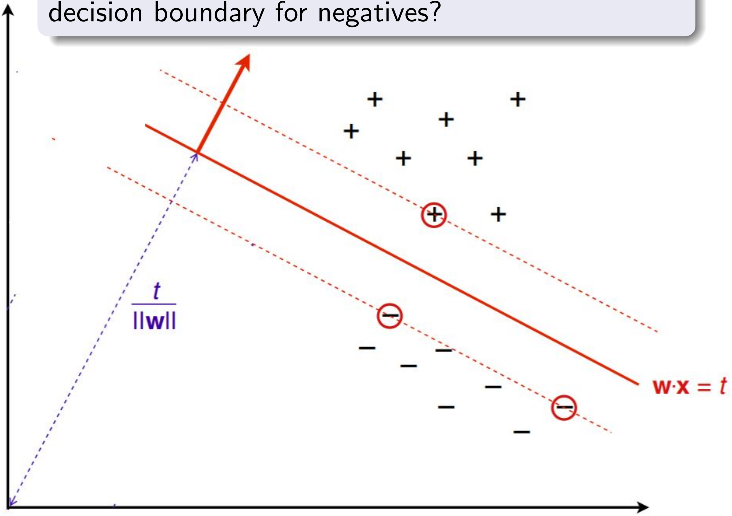


Calculating the margin



Calculating the margin

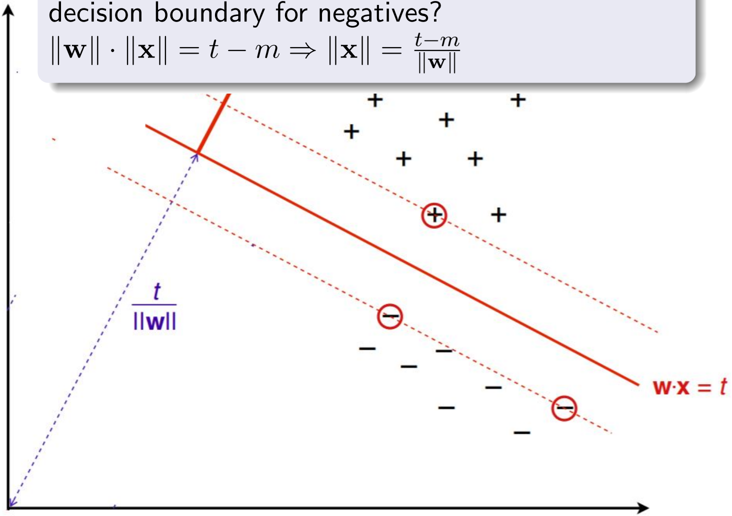
What is the distance between origin and the decision boundary for negatives?



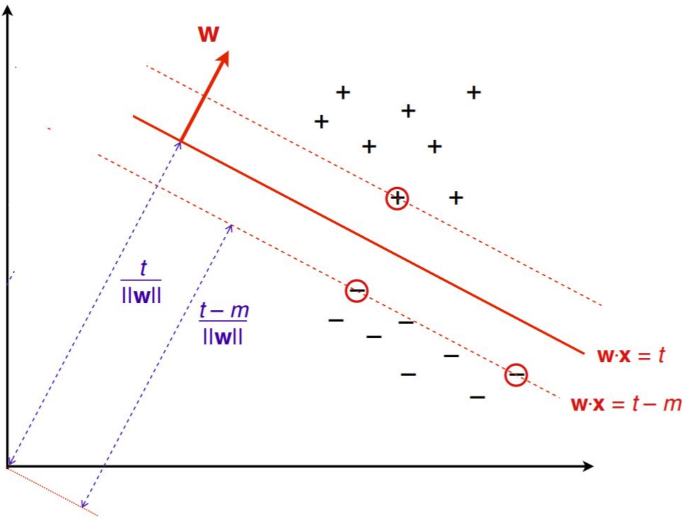
Calculating the margin

What is the distance between origin and the decision boundary for negatives?

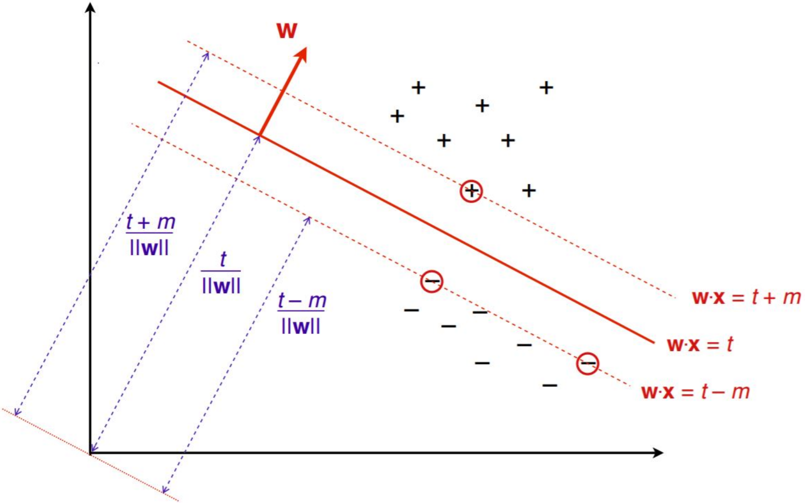
$\|\mathbf{w}\| \cdot \|\mathbf{x}\| = t - m \Rightarrow \|\mathbf{x}\| = \frac{t-m}{\|\mathbf{w}\|}$



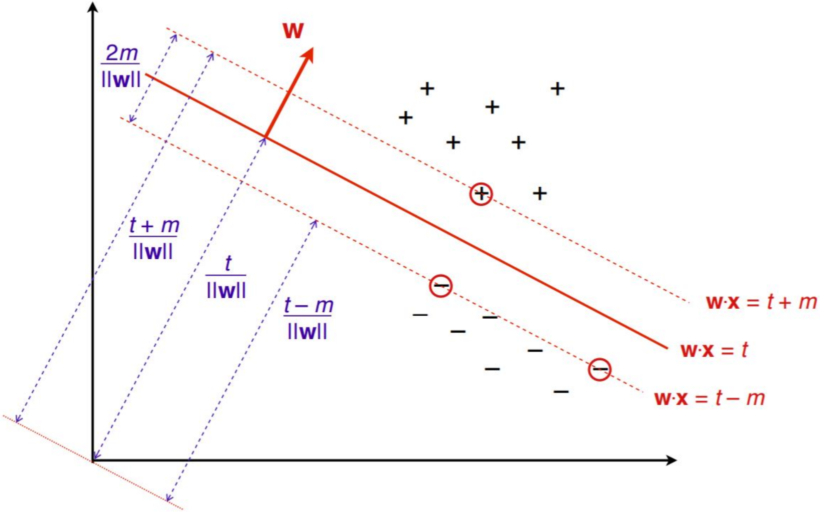
Calculating the margin



Calculating the margin

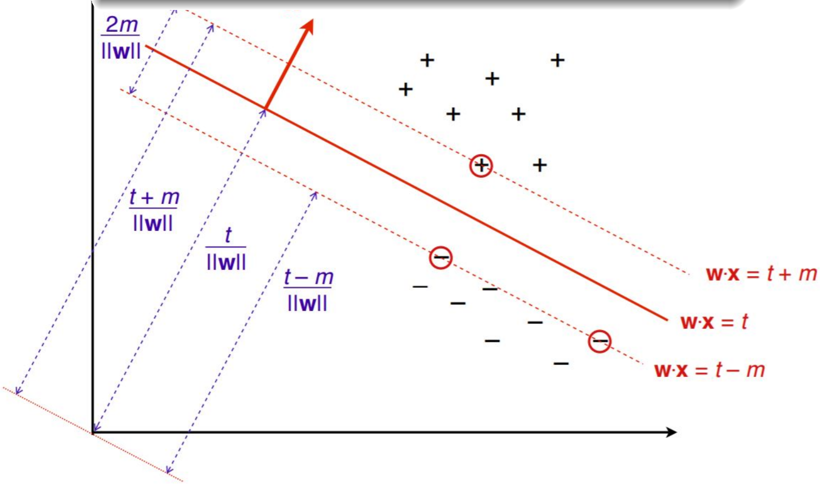


Key distances in SVM



Key distances in SVM

Since we are free to rescale t, $\mathbf{w}$ and m, it is customary to choose $m = 1$.



Optimization task in SVM

- Maximize the margin $\frac{1}{\|\mathbf{w}\|}$ such that
  - positives are at least by margin above the decision boundary: $\mathbf{w} \cdot \mathbf{x}_i \geq t + 1$
  - negatives are at least by margin below the decision boundary: $\mathbf{w} \cdot \mathbf{x}_i \leq t - 1$
- More conveniently and equivalently:

$$\mathbf{w}^*, t^* = \operatorname*{argmin}_{\mathbf{w},t} \frac{1}{2}\|\mathbf{w}\|^2$$

subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i - t) \geq 1, \quad y_i \in \{-1, 1\}, \quad 1 \leq i \leq n$

Optimization task in SVM

- Maximize the margin $\frac{1}{\|\mathbf{w}\|}$ such that
  - positives are at least by margin above the decision boundary: $\mathbf{w} \cdot \mathbf{x}_i \geq t + 1$
  - negatives are at least by margin below the decision boundary: $\mathbf{w} \cdot \mathbf{x}_i \leq t - 1$
- More conveniently and equivalently:

$$\mathbf{w}^*, t^* = \underset{\mathbf{w}, t}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2$$

subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i - t) \geq 1, \quad y_i \in \{-1, 1\}, \quad 1 \leq i \leq n$

Suppose $y_i(\mathbf{w} \cdot \mathbf{x}_i - t) = 1$. What can we say about $\mathbf{x}_i$?

Hard-margin SVM

$$\mathbf{w}^*, t^* = \underset{\mathbf{w},t}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2$$

subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i - t) \geq 1, \quad y_i \in \{-1, 1\}, \quad 1 \leq i \leq n$

To solve this optimization problem, first let's form the Lagrange function

$$\Lambda(\mathbf{w}, t, \alpha_1, \ldots, \alpha_n) =$$

Hard-margin SVM

$$\mathbf{w}^*, t^* = \underset{\mathbf{w},t}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2$$

subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i - t) \geq 1, \quad y_i \in \{-1, 1\}, \quad 1 \leq i \leq n$

To solve this optimization problem, first let's form the Lagrange function

$$\Lambda(\mathbf{w}, t, \alpha_1, \ldots, \alpha_n) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i (y_i(\mathbf{w} \cdot \mathbf{x}_i - t) - 1) =$$

Hard-margin SVM

$$\mathbf{w}^*, t^* = \underset{\mathbf{w},t}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2$$

subject to $y_i(\mathbf{w}\cdot\mathbf{x}_i - t) \geq 1, \quad y_i \in \{-1, 1\}, \quad 1 \leq i \leq n$

To solve this optimization problem, first let's form the Lagrange function

$$\Lambda(\mathbf{w}, t, \alpha_1, \ldots, \alpha_n) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i(y_i(\mathbf{w}\cdot\mathbf{x}_i - t) - 1) =$$

$$= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i y_i(\mathbf{w}\cdot\mathbf{x}_i) + \sum_{i=1}^{n} \alpha_i y_i t + \sum_{i=1}^{n} \alpha_i =$$

Hard-margin SVM

$$\mathbf{w}^*, t^* = \underset{\mathbf{w},t}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2$$

subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i - t) \geq 1, \quad y_i \in \{-1, 1\}, \quad 1 \leq i \leq n$

To solve this optimization problem, first let's form the Lagrange function

$$\begin{aligned}
\Lambda(\mathbf{w}, t, \alpha_1, \ldots, \alpha_n) &= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i (y_i(\mathbf{w} \cdot \mathbf{x}_i - t) - 1) = \\
&= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i y_i (\mathbf{w} \cdot \mathbf{x}_i) + \sum_{i=1}^{n} \alpha_i y_i t + \sum_{i=1}^{n} \alpha_i = \\
&= \frac{1}{2}\mathbf{w} \cdot \mathbf{w} - \mathbf{w} \cdot \Big(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i\Big) + t\Big(\sum_{i=1}^{n} \alpha_i y_i\Big) + \sum_{i=1}^{n} \alpha_i
\end{aligned}$$

Gradients of the Lagragian

$$\Lambda(\mathbf{w}, t, \alpha_1, \ldots, \alpha_n) = \frac{1}{2}\mathbf{w} \cdot \mathbf{w} - \mathbf{w} \cdot \left(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \right) + t \left(\sum_{i=1}^{n} \alpha_i y_i \right) + \sum_{i=1}^{n} \alpha_i$$

Gradients of the Lagragian

$$\Lambda(\mathbf{w}, t, \alpha_1, \ldots, \alpha_n) = \frac{1}{2}\mathbf{w} \cdot \mathbf{w} - \mathbf{w} \cdot \left(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i\right) + t\left(\sum_{i=1}^{n} \alpha_i y_i\right) + \sum_{i=1}^{n} \alpha_i$$

$$\frac{\partial}{\partial t}\Lambda(\mathbf{w}, t, \alpha_1, \ldots, \alpha_n) = \sum_{i=1}^{n} \alpha_i y_i = 0$$

Gradients of the Lagragian

$$\Lambda(\mathbf{w}, t, \alpha_1, \ldots, \alpha_n) = \frac{1}{2}\mathbf{w} \cdot \mathbf{w} - \mathbf{w} \cdot \left(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \right) + t \left(\sum_{i=1}^{n} \alpha_i y_i \right) + \sum_{i=1}^{n} \alpha_i$$

$$\frac{\partial}{\partial t} \Lambda(\mathbf{w}, t, \alpha_1, \ldots, \alpha_n) = \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\frac{\partial}{\partial \mathbf{w}} \Lambda(\mathbf{w}, t, \alpha_1, \ldots, \alpha_n) = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0$$

Gradients of the Lagragian

$$\Lambda(\mathbf{w}, t, \alpha_1, \ldots, \alpha_n) = \frac{1}{2}\mathbf{w} \cdot \mathbf{w} - \mathbf{w} \cdot \left(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \right) + t\left(\sum_{i=1}^{n} \alpha_i y_i \right) + \sum_{i=1}^{n} \alpha_i$$

$$\frac{\partial}{\partial t}\Lambda(\mathbf{w}, t, \alpha_1, \ldots, \alpha_n) = \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\frac{\partial}{\partial \mathbf{w}}\Lambda(\mathbf{w}, t, \alpha_1, \ldots, \alpha_n) = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0$$

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

Plugging back into the Lagragian

$$\Lambda(\mathbf{w}, t, \alpha_1, \ldots, \alpha_n) = \frac{1}{2}\mathbf{w} \cdot \mathbf{w} - \mathbf{w} \cdot \left(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i\right) + t\left(\sum_{i=1}^{n} \alpha_i y_i\right) + \sum_{i=1}^{n} \alpha_i$$

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

Plugging back into the Lagragian

$$\Lambda(\mathbf{w}, t, \alpha_1, \ldots, \alpha_n) = \frac{1}{2}\mathbf{w}\cdot\mathbf{w} - \mathbf{w}\cdot\Big(\sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i\Big) + t\Big(\sum_{i=1}^{n}\alpha_i y_i\Big) + \sum_{i=1}^{n}\alpha_i$$

$$\mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i \quad \sum_{i=1}^{n}\alpha_i y_i = 0$$

$$\Lambda(\alpha_1, \ldots, \alpha_n) = -\frac{1}{2}\Big(\sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i\Big)\cdot\Big(\sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i\Big) + \sum_{i=1}^{n}\alpha_i =$$

Plugging back into the Lagrangian

$$\Lambda(\mathbf{w}, t, \alpha_1, \ldots, \alpha_n) = \frac{1}{2}\mathbf{w} \cdot \mathbf{w} - \mathbf{w} \cdot \Big(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i\Big) + t\Big(\sum_{i=1}^{n} \alpha_i y_i\Big) + \sum_{i=1}^{n} \alpha_i$$

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\Lambda(\alpha_1, \ldots, \alpha_n) = -\frac{1}{2}\Big(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i\Big) \cdot \Big(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i\Big) + \sum_{i=1}^{n} \alpha_i =$$

$$= -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j + \sum_{i=1}^{n} \alpha_i$$

Applying KKT method on the Lagragian

$$\Lambda(\alpha_1, \ldots, \alpha_n) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j + \sum_{i=1}^{n}\alpha_i$$

$$\alpha_1^*, \ldots, \alpha_n^* = \underset{\alpha_1,\ldots,\alpha_n}{\operatorname{argmax}} -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j + \sum_{i=1}^{n}\alpha_i$$

$$\text{subject to } \alpha_i \geq 0, \quad i = 1, \ldots, n \text{ and } \sum_{i=1}^{n}\alpha_i y_i = 0$$

This task can be solved by quadratic optimisation solvers as we will see during the lab session.

Summary of SVM optimization

$$\mathbf{w}^*, t^* = \underset{\mathbf{w},t}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2$$

subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i - t) \geq 1, \quad y_i \in \{-1, 1\}, \quad i = 1, \ldots, n$

Summary of SVM optimization

$$\mathbf{w}^*, t^* = \underset{\mathbf{w},t}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2$$

subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i - t) \geq 1, \quad y_i \in \{-1, 1\}, \quad i = 1, \dots, n$

We can solve the equivalent dual problem

$$\alpha_1^*, \dots, \alpha_n^* = \underset{\alpha_1,\dots,\alpha_n}{\operatorname{argmax}} -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j + \sum_{i=1}^{n}\alpha_i$$

subject to $\alpha_i \geq 0, \quad i = 1, \dots, n$ and $\sum_{i=1}^{n}\alpha_i y_i = 0$

Summary of SVM optimization

$$\mathbf{w}^*, t^* = \underset{\mathbf{w},t}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2$$

subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i - t) \geq 1, \quad y_i \in \{-1, 1\}, \quad i = 1, \ldots, n$

We can solve the equivalent dual problem

$$\alpha_1^*, \ldots, \alpha_n^* = \underset{\alpha_1,\ldots,\alpha_n}{\operatorname{argmax}} -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j + \sum_{i=1}^{n} \alpha_i$$

subject to $\alpha_i \geq 0, \quad i = 1, \ldots, n$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$

From the result we can calculate:

$$\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i \mathbf{x}_i \qquad t^* = \mathbf{w}^* \cdot \mathbf{x}_i - y_i,$$

where $\mathbf{x}_i$ is a support vector and α_i is its weight.

Hard margin and soft margin SVM

- This was a **hard margin SVM** – we assumed that the classes are linearly separable

Hard margin and soft margin SVM

- This was a **hard margin SVM** – we assumed that the classes are linearly separable
- **Soft margin SVM** can tolerate **margin errors**: cases where an instance is inside the margin or even at the wrong side of the decision boundary

Hard margin and soft margin SVM

- This was a **hard margin SVM** – we assumed that the classes are linearly separable
- **Soft margin SVM** can tolerate **margin errors**: cases where an instance is inside the margin or even at the wrong side of the decision boundary
- The idea is to introduce slack variables $\xi_i \geq 0$, one for each instance, measuring the amount of margin error (or equal to 0 if no error)

Soft margin SVM

- The task is the following:

$$\mathbf{w}^*, t^*, \xi_i^* = \operatorname*{argmin}_{\mathbf{w}, t, \xi_i} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

subject to $y_i(\mathbf{w}\cdot\mathbf{x}_i - t) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n$

Soft margin SVM

- The task is the following:

$$\mathbf{w}^*, t^*, \xi_i^* = \underset{\mathbf{w}, t, \xi_i}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{n} \xi_i$$

subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i - t) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n$

- Here C is a regularisation parameter:

Soft margin SVM

- The task is the following:

$$\mathbf{w}^*, t^*, \xi_i^* = \underset{\mathbf{w}, t, \xi_i}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i$$

subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i - t) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n$

- Here C is a regularisation parameter:
  - Higher C means more penalty on margin errors

Soft margin SVM

- The task is the following:

$$\mathbf{w}^*, t^*, \xi_i^* = \underset{\mathbf{w}, t, \xi_i}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{n} \xi_i$$

subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i - t) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n$

- Here C is a regularisation parameter:
  - Higher C means more penalty on margin errors
  - Lower C means less penalty on margin errors

Soft margin SVM

- The task is the following:

$$\mathbf{w}^*, t^*, \xi_i^* = \underset{\mathbf{w}, t, \xi_i}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{n} \xi_i$$

subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i - t) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n$

- Here C is a regularisation parameter:
  - Higher C means more penalty on margin errors
  - Lower C means less penalty on margin errors
- Higher C usually results in more support vectors, hence C is a **complexity parameter**

Lagragian function

$$\mathbf{w}^*, t^*, \xi_i^* = \operatorname*{argmin}_{\mathbf{w}, t, \xi_i} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

subject to $y_i(\mathbf{w}\cdot\mathbf{x}_i - t) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n$

$$\Lambda(\mathbf{w}, t, , \xi_i, \alpha_i, \beta_i) =$$

Lagragian function

$$\mathbf{w}^*, t^*, \xi_i^* = \operatorname*{argmin}_{\mathbf{w}, t, \xi_i} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i$$

subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i - t) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n$

$$\Lambda(\mathbf{w}, t, , \xi_i, \alpha_i, \beta_i) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i -$$

$$- \sum_{i=1}^{n} \alpha_i(y_i(\mathbf{w} \cdot \mathbf{x}_i - t) - (1 - \xi_i)) - \sum_{i=1}^{n} \beta_i \xi_i =$$

Lagragian function

$$\mathbf{w}^*, t^*, \xi_i^* = \underset{\mathbf{w}, t, \xi_i}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

subject to $y_i(\mathbf{w}\cdot\mathbf{x}_i - t) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n$

$$\begin{aligned}\Lambda(\mathbf{w}, t, , \xi_i, \alpha_i, \beta_i) = {} & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i - \\ & -\sum_{i=1}^{n}\alpha_i(y_i(\mathbf{w}\cdot\mathbf{x}_i - t) - (1 - \xi_i)) - \sum_{i=1}^{n}\beta_i\xi_i = \\ & = \Lambda(\mathbf{w}, t, \alpha_i) + \sum_{i=1}^{n}(C - \alpha_i - \beta_i)\xi_i\end{aligned}$$

Lagragian function

$$\mathbf{w}^*, t^*, \xi_i^* = \underset{\mathbf{w}, t, \xi_i}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

subject to $y_i(\mathbf{w}\cdot\mathbf{x}_i - t) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n$

$$\Lambda(\mathbf{w}, t, , \xi_i, \alpha_i, \beta_i) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i -$$

$$-\sum_{i=1}^{n}\alpha_i(y_i(\mathbf{w}\cdot\mathbf{x}_i - t) - (1 - \xi_i)) - \sum_{i=1}^{n}\beta_i\xi_i =$$

$$= \Lambda(\mathbf{w}, t, \alpha_i) + \sum_{i=1}^{n}(C - \alpha_i - \beta_i)\xi_i$$

$$\frac{\partial}{\partial \xi_{\mathbf{i}}}\Lambda(\mathbf{w}, t, , \xi_i, \alpha_i, \beta_i) =$$

Lagragian function

$$\mathbf{w}^*, t^*, \xi_i^* = \underset{\mathbf{w},t,\xi_i}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

subject to $y_i(\mathbf{w}\cdot\mathbf{x}_i - t) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n$

$$\begin{aligned}\Lambda(\mathbf{w}, t, , \xi_i, \alpha_i, \beta_i) &= \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i - \\ &- \sum_{i=1}^{n}\alpha_i(y_i(\mathbf{w}\cdot\mathbf{x}_i - t) - (1 - \xi_i)) - \sum_{i=1}^{n}\beta_i\xi_i = \\ &= \Lambda(\mathbf{w}, t, \alpha_i) + \sum_{i=1}^{n}(C - \alpha_i - \beta_i)\xi_i\end{aligned}$$

$$\frac{\partial}{\partial\xi_{\mathbf{i}}}\Lambda(\mathbf{w}, t, , \xi_i, \alpha_i, \beta_i) = C - \alpha_i - \beta_i = 0$$

Lagragian function

$$\mathbf{w}^*, t^*, \xi_i^* = \operatorname*{argmin}_{\mathbf{w},t,\xi_i} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

subject to $y_i(\mathbf{w}\cdot\mathbf{x}_i - t) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n$

$$\Lambda(\mathbf{w}, t, , \xi_i, \alpha_i, \beta_i) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i -$$

$$-\sum_{i=1}^{n}\alpha_i(y_i(\mathbf{w}\cdot\mathbf{x}_i - t) - (1 - \xi_i)) - \sum_{i=1}^{n}\beta_i\xi_i =$$

$$= \Lambda(\mathbf{w}, t, \alpha_i) + \sum_{i=1}^{n}(C - \alpha_i - \beta_i)\xi_i$$

$$\frac{\partial}{\partial \xi_\mathbf{i}}\Lambda(\mathbf{w}, t, , \xi_i, \alpha_i, \beta_i) = C - \alpha_i - \beta_i = 0 \Rightarrow \alpha_i \leq C,$$

since $\beta_i \geq 0$

Summary of soft-margin SVM

$$\mathbf{w}^*, t^*, \xi_i^* = \underset{\mathbf{w}, t, \xi_i}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i - t) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n$

Summary of soft-margin SVM

$$\mathbf{w}^*, t^*, \xi_i^* = \underset{\mathbf{w}, t, \xi_i}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i$$

subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i - t) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n$

We can solve the equivalent dual problem

$$\alpha_1^*, \ldots, \alpha_n^* = \underset{\alpha_1, \ldots, \alpha_n}{\operatorname{argmax}} -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j + \sum_{i=1}^{n} \alpha_i$$

subject to $0 \leq \alpha_i \leq C, \quad i = 1, \ldots, n$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$

Summary of soft-margin SVM

$$\mathbf{w}^*, t^*, \xi_i^* = \operatorname*{argmin}_{\mathbf{w},t,\xi_i} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

subject to $y_i(\mathbf{w}\cdot\mathbf{x}_i - t) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n$

We can solve the equivalent dual problem

$$\alpha_1^*, \ldots, \alpha_n^* = \operatorname*{argmax}_{\alpha_1,\ldots,\alpha_n} -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j + \sum_{i=1}^{n}\alpha_i$$

subject to $0 \leq \alpha_i \leq C, \quad i = 1, \ldots, n$ and $\sum_{i=1}^{n}\alpha_i y_i = 0$

From the result we can calculate:

$$\mathbf{w}^* = \sum_{i=1}^{n}\alpha_i^* y_i \mathbf{x}_i \qquad t^* = \mathbf{w}^*\cdot\mathbf{x}_i - y_i,$$

where $\mathbf{x}_i$ is a support vector and α_i is its weight.

Soft-margin SVM

Pro: usually **generalizes** better than the other discussed linear methods due to margin maximization

Soft-margin SVM

Pro: usually **generalizes** better than the other discussed linear methods due to margin maximization

Pro: can be used with kernels (discussed soon)

Soft-margin SVM

Pro: usually **generalizes** better than the other discussed linear methods due to margin maximization

Pro: can be used with kernels (discussed soon)

Pro: works even if the classes are not linearly separable

Soft-margin SVM

Pro: usually **generalizes** better than the other discussed linear methods due to margin maximization

Pro: can be used with kernels (discussed soon)

Pro: works even if the classes are not linearly separable

Con: the fitted model depends on very few instances (support vectors) and ignores the location of other points (as long as they are on the correct side of the decision boundary)

Soft-margin SVM

Pro: usually **generalizes** better than the other discussed linear methods due to margin maximization

Pro: can be used with kernels (discussed soon)

Pro: works even if the classes are not linearly separable

Con: the fitted model depends on very few instances (support vectors) and ignores the location of other points (as long as they are on the correct side of the decision boundary)

Con: Works efficiently on relatively small datasets

What have we learned today?

- ✓ Convex Optimization
- ✓ Support Vector Machine (SVM)
- ✓ Hard-margin SVM
- ✓ Soft-margin SVM