2nd feb 2023

Lecture 3

Tian Han

Outline

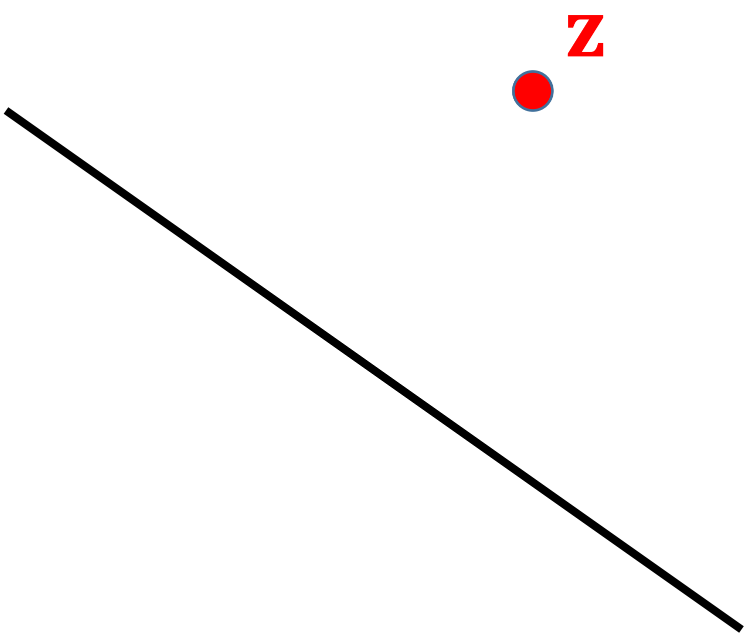
- Support Vector Machine (SVM)
- Regularization
- Convex optimization basics

Support Vector Machine (SVM)

Project a Point onto a Hyperplane

Project a Point onto a Hyperplane

Question: how to project $\mathbf{z}$ onto the hyperplane?

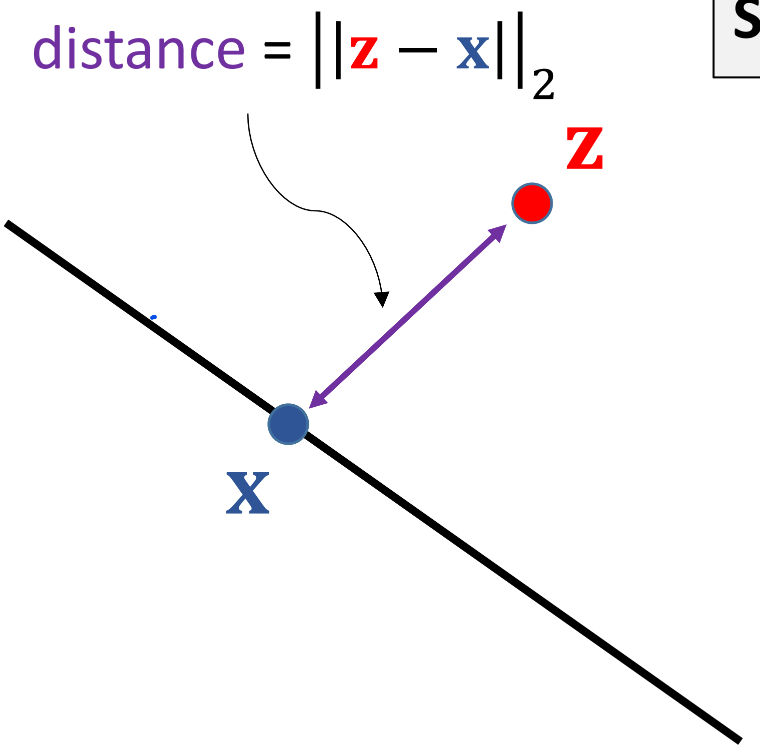


Hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$

Project a Point onto a Hyperplane

Question: how to project $\mathbf{z}$ onto the hyperplane?

Solution: find $\mathbf{x}$ on the hyperplane such that $||\mathbf{z} - \mathbf{x}||_2^2$ is minimized.



Hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$

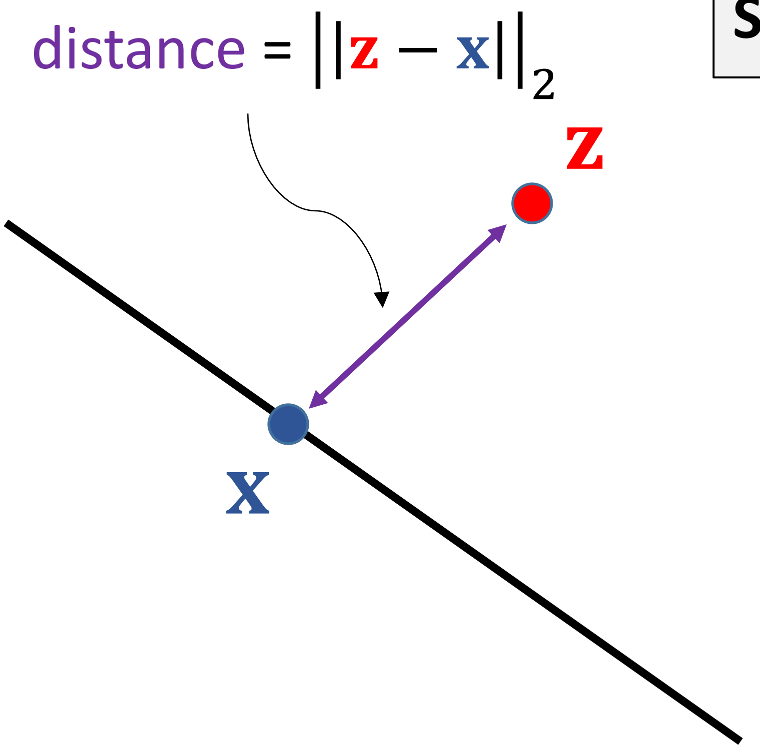
- $\min_{\mathbf{x}} ||\mathbf{z} - \mathbf{x}||_2^2; \quad$ s.t. $\mathbf{w}^T\mathbf{x} + b = 0$

x is arbitrary points on the hyperplane
and are trying to minimize the distance between z and x to
get the projection on to the hyperplane

Project a Point onto a Hyperplane

Question: how to project $\mathbf{z}$ onto the hyperplane?

Solution: find $\mathbf{x}$ on the hyperplane such that $||\mathbf{z}-\mathbf{x}||_2^2$ is minimized.



Hyperplane $\mathbf{w}^T\mathbf{x}+b=0$

- $\min_{\mathbf{x}} ||\mathbf{z}-\mathbf{x}||_2^2; \quad \text{s.t. } \mathbf{w}^T\mathbf{x}+b=0$
- Solve the problem using the Lagrange multiplier:

$$\begin{cases} \dfrac{\partial ||\mathbf{z}-\mathbf{x}||_2^2}{\partial \mathbf{x}} + \lambda \dfrac{\partial\,(\mathbf{w}^T\mathbf{x}+b)}{\partial \mathbf{x}} = 0; \\ \mathbf{w}^T\mathbf{x}+b=0. \end{cases}$$

- Solution: $\boxed{\mathbf{x} = \mathbf{z} - \dfrac{\mathbf{w}^T\mathbf{z}+b}{||\mathbf{w}||_2^2}\mathbf{w}}$

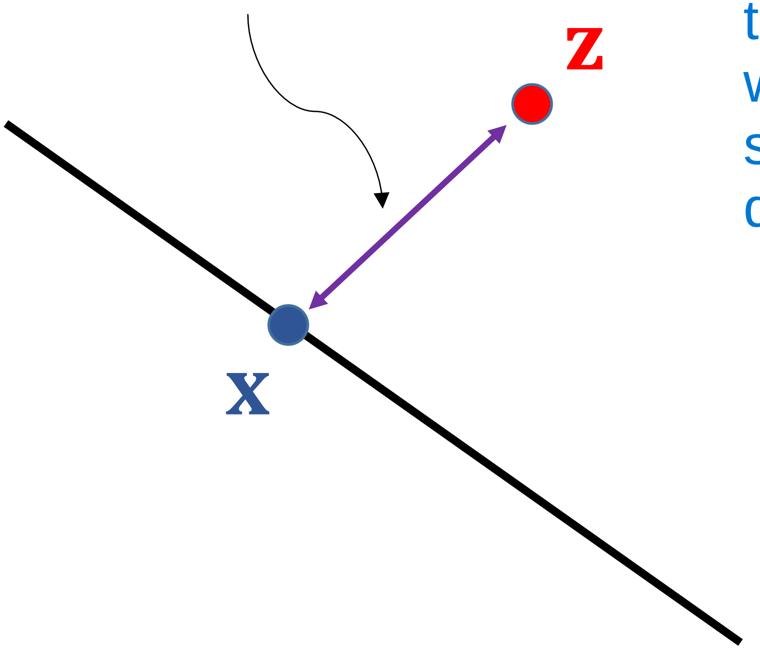
Projection

Project a Point onto a Hyperplane

Question: how to project $\mathbf{z}$ onto the hyperplane?

Solution: find $\mathbf{x}$ on the hyperplane such that $||\mathbf{z} - \mathbf{x}||_2^2$ is minimized.

distance = $||\mathbf{z} - \mathbf{x}||_2$



Hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$

equation

this is the x with the smallest distance

- Solution: $\mathbf{x} = \mathbf{z} - \frac{\mathbf{w}^T\mathbf{z}+b}{||\mathbf{w}||_2^2}\mathbf{w}$
- The ℓ_2 distance between $\mathbf{z}$ and the hyperplane is

$$\boxed{||\mathbf{z} - \mathbf{x}||_2 = \frac{|\mathbf{w}^T\mathbf{z}+b|}{||\mathbf{w}||_2}.}$$

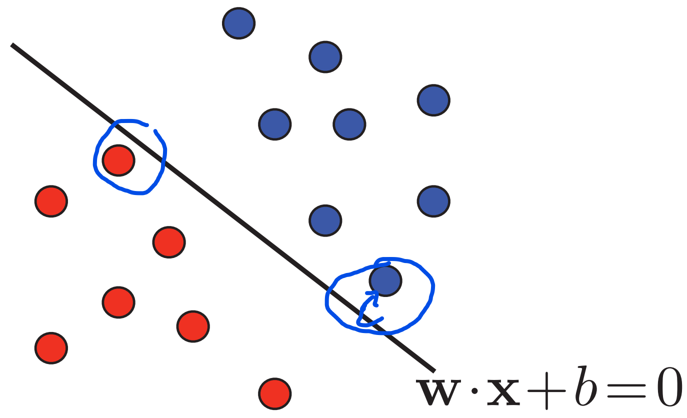
w is the model weights

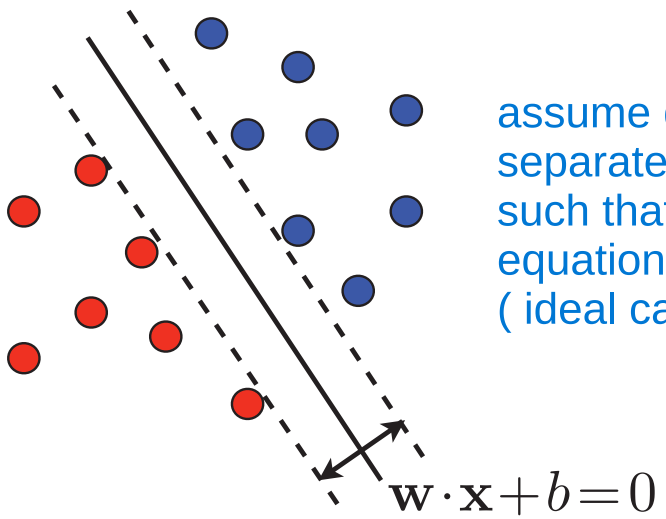
b is the bias

Support Vector Machine (SVM)

Support Vector Machine (SVM)

Separate data by a hyperplane (assume the data are separable)





assume data can be linearly separated
such that we can find an equation to separate the data
(ideal case)

An arbitrary hyperplane.

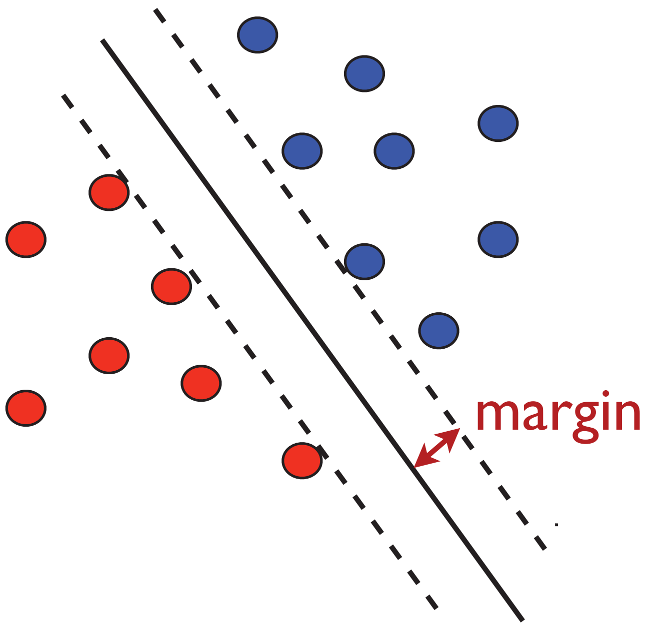
The hyperplane that maximizes the margin.

within this margin there are no training points

The figure is from the book "*Foundations of Machine Learning*"

Support Vector Machine (SVM)

Separate data by a hyperplane (assume the data are separable)



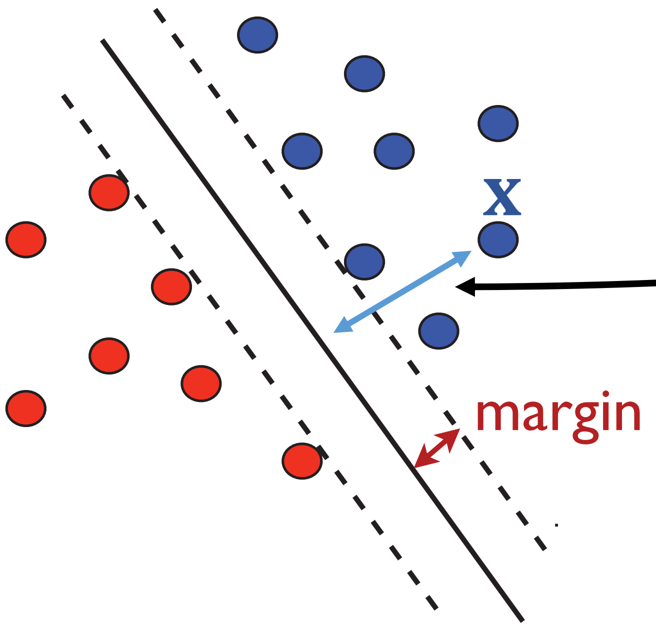
Hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$

it can be called as minimum distance of training data points from the classifier hyperplane
this can be computed for every training sample

The figure is from the book "*Foundations of Machine Learning*"

Support Vector Machine (SVM)

Separate data by a hyperplane (assume the data are separable)



Hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$

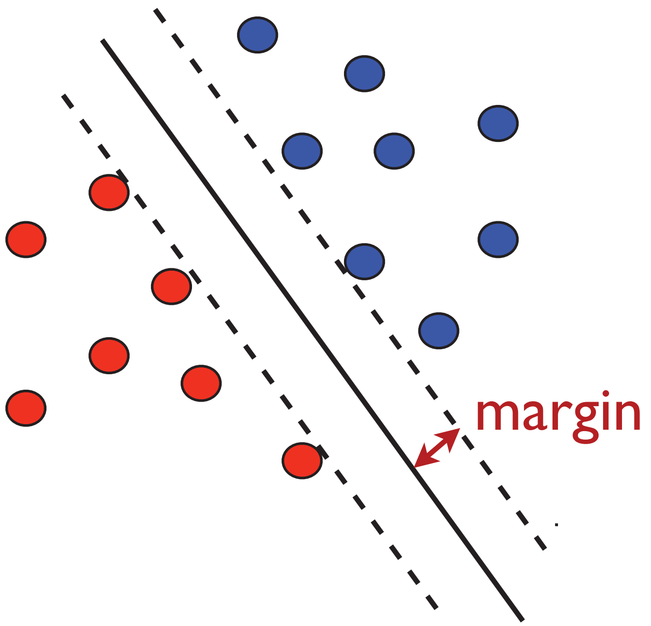
- The distance between any feature vector, $\mathbf{x}$, and the hyperplane is

$$\boxed{\text{dist} = \frac{|\mathbf{w}^T\mathbf{x}+b|}{\|\mathbf{w}\|_2}.}$$

The figure is from the book "*Foundations of Machine Learning*"

Support Vector Machine (SVM)

Separate data by a hyperplane (assume the data are separable)



Hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$

- The distance between any feature vector, $\mathbf{x}$, and the hyperplane is

$$\text{dist} = \frac{|\mathbf{w}^T\mathbf{x}+b|}{||\mathbf{w}||_2}.$$

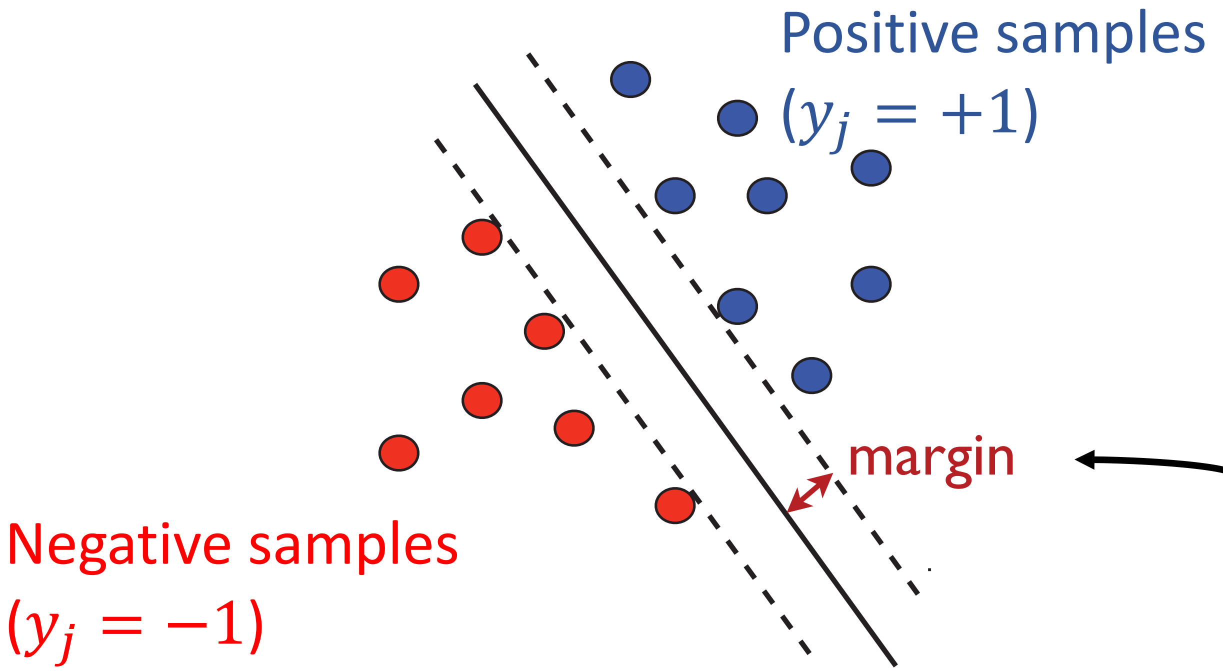
- The margin is the smallest distance:

$$\min_j \frac{|\mathbf{w}^T\mathbf{x}_j+b|}{||\mathbf{w}||_2}$$

The figure is from the book "*Foundations of Machine Learning*"

Support Vector Machine (SVM)

Separate data by a hyperplane (assume the data are separable)



Hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$

> 0 for positive class
< 0 for negative class

- The distance between any feature vector, $\mathbf{x}$, and the hyperplane is

$$\text{dist} = \frac{|\mathbf{w}^T\mathbf{x}+b|}{||\mathbf{w}||_2}.$$

- The margin is the smallest distance:

$$\min_j \frac{|\mathbf{w}^T\mathbf{x}_j+b|}{||\mathbf{w}||_2} = \min_j \frac{y_j(\mathbf{w}^T\mathbf{x}_j+b)}{||\mathbf{w}||_2}$$

+ve for positive class as both positive
+ve for negative class as well as both negative
hence for correct prediction it is +ve

The figure is from the book "*Foundations of Machine Learning*"

Support Vector Machine (SVM)

it is the multiplication between the target value label and its response divided by the weight

Margin = $\min_{j} \frac{y_j(\mathbf{w}^T\mathbf{x}_j + b)}{||\mathbf{w}||_2}$; we want to maximize the margin.

Support Vector Machine (SVM)

Margin = $\min_j \frac{y_j \boxed{(\mathbf{w}^T \mathbf{x}_j + b)}}{||\mathbf{w}||_2}$; we want to maximize the margin.

Define $\bar{\mathbf{x}}_j = [\mathbf{x}_j; 1] \in \mathbb{R}^{d+1}$

Define $\bar{\mathbf{w}} = [\mathbf{w}, b] \in \mathbb{R}^{d+1}$

➔ $\boxed{\mathbf{x}_j^T \mathbf{w} + b = \bar{\mathbf{x}}_j^T \bar{\mathbf{w}}}$

Support Vector Machine (SVM)

Margin = $\min_j \frac{y_j \mathbf{w}^T \mathbf{x}_j}{||\mathbf{w}||_2}$; we want to maximize the margin.

maximise the weights with the margin

minimise while going over all model samples

Support Vector Machine (SVM): $\max_{\mathbf{w}} \min_j \frac{y_j \mathbf{w}^T \mathbf{x}_j}{||\mathbf{w}||_2}$

Support Vector Machine (SVM)

Support Vector Machine (SVM): $$\max_{\mathbf{w}} \min_{j} \frac{y_j \mathbf{w}^T \mathbf{x}_j}{||\mathbf{w}||_2}$$

Support Vector Machine (SVM)

Support Vector Machine (SVM): $\max_{\mathbf{w}} \min_{j} \frac{y_j \mathbf{w}^T \mathbf{x}_j}{||\mathbf{w}||_2}$

$$\underset{\mathbf{w}}{\operatorname{argmax}} \min_{j} \frac{y_j \mathbf{w}^T \mathbf{x}_j}{||\mathbf{w}||_2} = \underset{\mathbf{w}}{\operatorname{argmax}} \frac{\min_{j} y_j \mathbf{w}^T \mathbf{x}_j}{||\mathbf{w}||_2}$$

Support Vector Machine (SVM)

Support Vector Machine (SVM): $\max_{\mathbf{w}} \min_{j} \frac{y_j \mathbf{w}^T \mathbf{x}_j}{||\mathbf{w}||_2}$

$$\operatorname*{argmax}_{\mathbf{w}} \min_{j} \frac{y_j \mathbf{w}^T \mathbf{x}_j}{||\mathbf{w}||_2} = \operatorname*{argmax}_{\mathbf{w}} \frac{\min_{j} y_j \mathbf{w}^T \mathbf{x}_j}{||\mathbf{w}||_2}$$

$$= \operatorname*{argmax}_{\mathbf{w}} \frac{1}{||\mathbf{w}||_2}, \qquad \text{s.t.} \quad \left(\min_{j} \; y_j \mathbf{w}^T \mathbf{x}_j\right) = 1$$

fix the numerator to do the maximisation of the other term

Support Vector Machine (SVM)

Support Vector Machine (SVM): $\max_{\mathbf{w}} \min_{j} \frac{y_j \mathbf{w}^T \mathbf{x}_j}{||\mathbf{w}||_2}$

$$\operatorname*{argmax}_{\mathbf{w}} \min_{j} \frac{y_j \mathbf{w}^T \mathbf{x}_j}{||\mathbf{w}||_2} = \operatorname*{argmax}_{\mathbf{w}} \frac{\min_{j} y_j \mathbf{w}^T \mathbf{x}_j}{||\mathbf{w}||_2}$$

$$= \operatorname*{argmax}_{\mathbf{w}} \frac{1}{||\mathbf{w}||_2}, \quad \text{s.t.} \quad \left(\min_{j} \ y_j \mathbf{w}^T \mathbf{x}_j\right) = 1$$

$$= \operatorname*{argmin}_{\mathbf{w}} ||\mathbf{w}||_2^2, \quad \text{s.t.} \quad \left(\min_{j} \ y_j \mathbf{w}^T \mathbf{x}_j\right) = 1$$

inverse so max becomes minimization

Support Vector Machine (SVM)

Support Vector Machine (SVM): $\max_{\mathbf{w}} \min_{j} \frac{y_j \mathbf{w}^T \mathbf{x}_j}{||\mathbf{w}||_2}$

$$\underset{\mathbf{w}}{\operatorname{argmax}} \min_{j} \frac{y_j \mathbf{w}^T \mathbf{x}_j}{||\mathbf{w}||_2} = \underset{\mathbf{w}}{\operatorname{argmax}} \frac{\min_{j} y_j \mathbf{w}^T \mathbf{x}_j}{||\mathbf{w}||_2}$$

$$= \underset{\mathbf{w}}{\operatorname{argmax}} \frac{1}{||\mathbf{w}||_2}, \quad \text{s.t.} \quad \left(\min_{j} \; y_j \mathbf{w}^T \mathbf{x}_j\right) = 1$$

$$= \underset{\mathbf{w}}{\operatorname{argmin}} \, ||\mathbf{w}||_2^2, \quad \text{s.t.} \quad \left(\min_{j} \; y_j \mathbf{w}^T \mathbf{x}_j\right) = 1$$

$$= \underset{\mathbf{w}}{\operatorname{argmin}} \, ||\mathbf{w}||_2^2, \quad \text{s.t.} \quad y_j \mathbf{w}^T \mathbf{x}_j \geq 1 \text{ for all } j$$

consider n data points a1 , a2.... an
if it is said that min of these points is 1
and lets say a1 is 1
this tells us that other points are greater than
or equal to 1

Support Vector Machine (SVM)

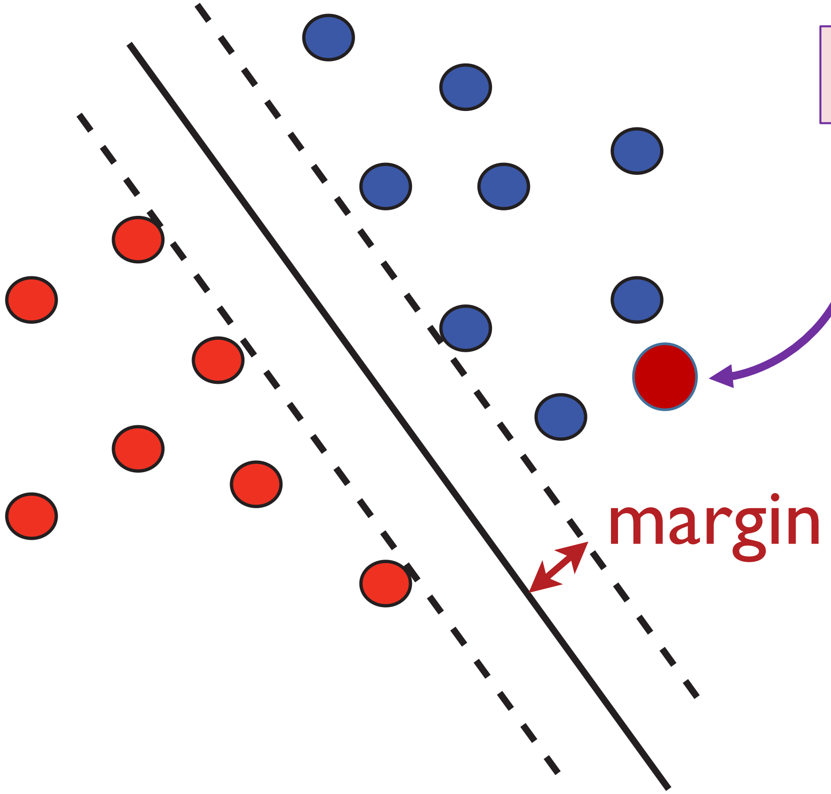
$$\min_{\mathbf{w}} ||\mathbf{w}||_2^2, \quad \text{s.t.} \quad y_j \mathbf{w}^T \mathbf{x}_j \geq 1 \text{ for all } j \in \{1, \cdots, n\}.$$

important assumption that data is linearly separable

Equivalent form of SVM

Support Vector Machine (SVM)

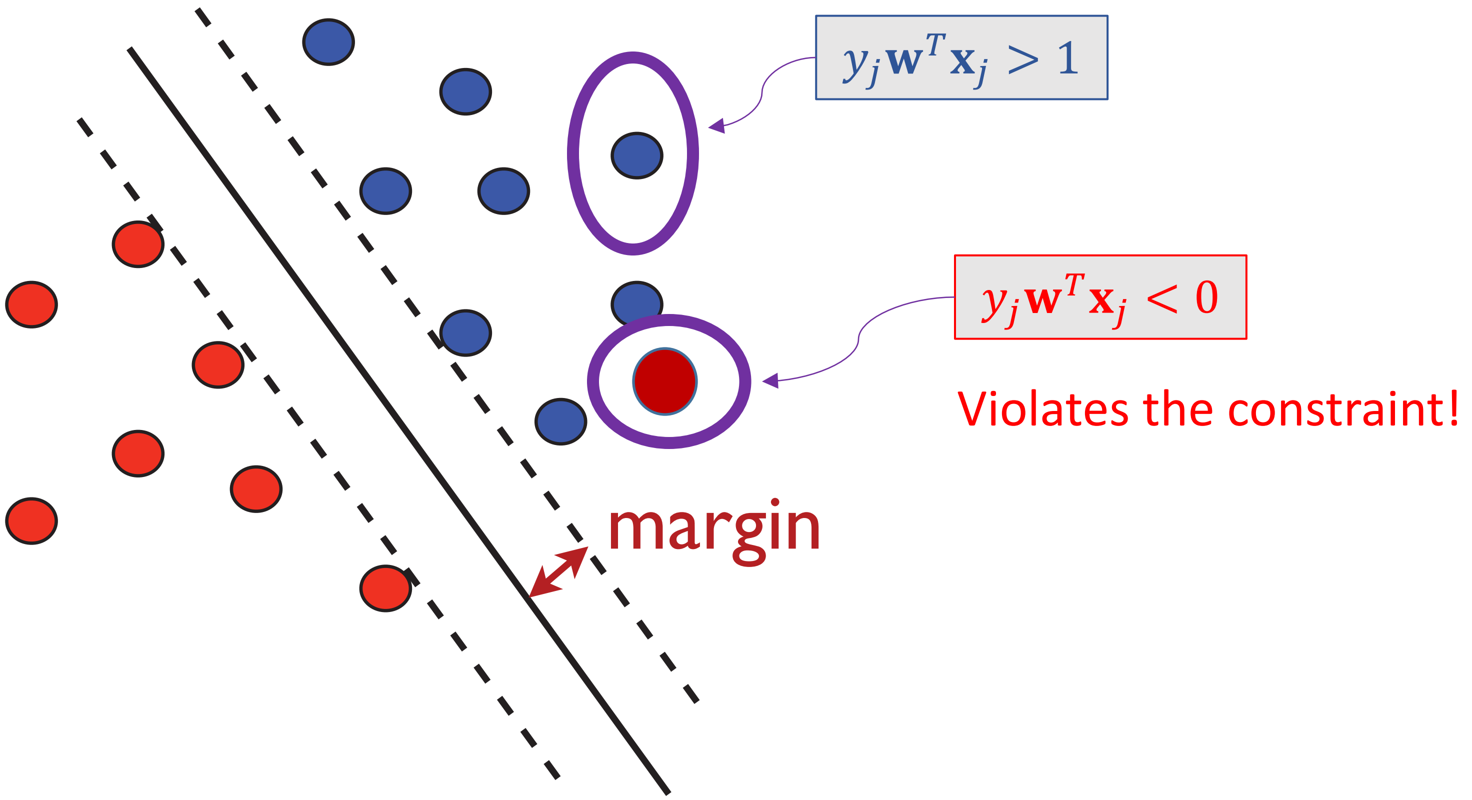
$$\min_{\mathbf{w}} ||\mathbf{w}||_2^2, \quad \text{s.t.} \quad y_j \mathbf{w}^T \mathbf{x}_j \geq 1 \text{ for all } j \in \{1, \cdots, n\}.$$



What if the data is inseparable?

Support Vector Machine (SVM)

$$\min_{\mathbf{w}} ||\mathbf{w}||_2^2, \quad \text{s.t.} \quad y_j \mathbf{w}^T \mathbf{x}_j \geq 1 \text{ for all } j \in \{1, \cdots, n\}.$$



Support Vector Machine (SVM)

$$\min_{\mathbf{w}} \|\mathbf{w}\|_2^2, \quad \text{s.t.} \quad 1 - y_j \mathbf{w}^T \mathbf{x}_j \leq 0 \ \text{ for all } j \in \{1, \cdots, n\}.$$

Relax

$$\min_{\mathbf{w}, \xi_j} \|\mathbf{w}\|_2^2 + \lambda \sum_j [\xi_j]_+, \quad \text{s.t.} \quad 1 - y_j \mathbf{w}^T \mathbf{x}_j = \xi_j \ \text{ for all } j \in \{1, \cdots, n\}.$$

- $[\xi_j]_+ = \max\{\xi_j, 0\}$

called slack variable which is also like a penalty for breaking the constraints when data isnt linearly separable

if you have negative value then u have original constraints

if its positive clearly it needs to be minimised with the new variable

Support Vector Machine (SVM)

$$\min_{\mathbf{w}} \|\mathbf{w}\|_2^2, \quad \text{s.t.} \quad 1 - y_j \mathbf{w}^T \mathbf{x}_j \leq 0 \ \text{ for all } j \in \{1, \cdots, n\}.$$

Relax

$$\min_{\mathbf{w}, \xi_j} \|\mathbf{w}\|_2^2 + \lambda \sum_j [\xi_j]_+, \quad \text{s.t.} \quad 1 - y_j \mathbf{w}^T \mathbf{x}_j = \xi_j \ \text{ for all } j \in \{1, \cdots, n\}.$$

- $[\xi_j]_+ = \max\{\xi_j, 0\}$
- $\xi_j \leq 0$ means the constraint $1 - y_j \mathbf{w}^T \mathbf{x}_j \leq 0$ is satisfied
  ➔ no penalty!
- $\xi_j > 0$ means the constraint is violated (because the data is inseparable)
  ➔ penalize the violation ξ_j.

mis classification gives larger penalty where as if the data point is within the margin but on the right class side then smaller penalty

kasai > 1 large penealty
0< kasai < 1 then small penalty

Support Vector Machine (SVM)

$$\min_{\mathbf{w}} ||\mathbf{w}||_2^2, \quad \text{s.t.} \quad 1 - y_j \mathbf{w}^T \mathbf{x}_j \leq 0 \;\; \text{for all } j \in \{1, \cdots, n\}.$$

$\Big\Downarrow$ **Relax**

objective function also given penalty

$$\min_{\mathbf{w}, \xi_j} ||\mathbf{w}||_2^2 + \lambda \sum_j [\xi_j]_+, \quad \text{s.t.} \quad 1 - y_j \mathbf{w}^T \mathbf{x}_j = \xi_j \;\; \text{for all } j \in \{1, \cdots, n\}.$$

$\Big\Updownarrow$ **Equivalent**

$$\min_{\mathbf{w}, b} ||\mathbf{w}||_2^2 + \lambda \sum_j [1 - y_j \mathbf{w}^T \mathbf{x}_j]_+.$$

Comparisons

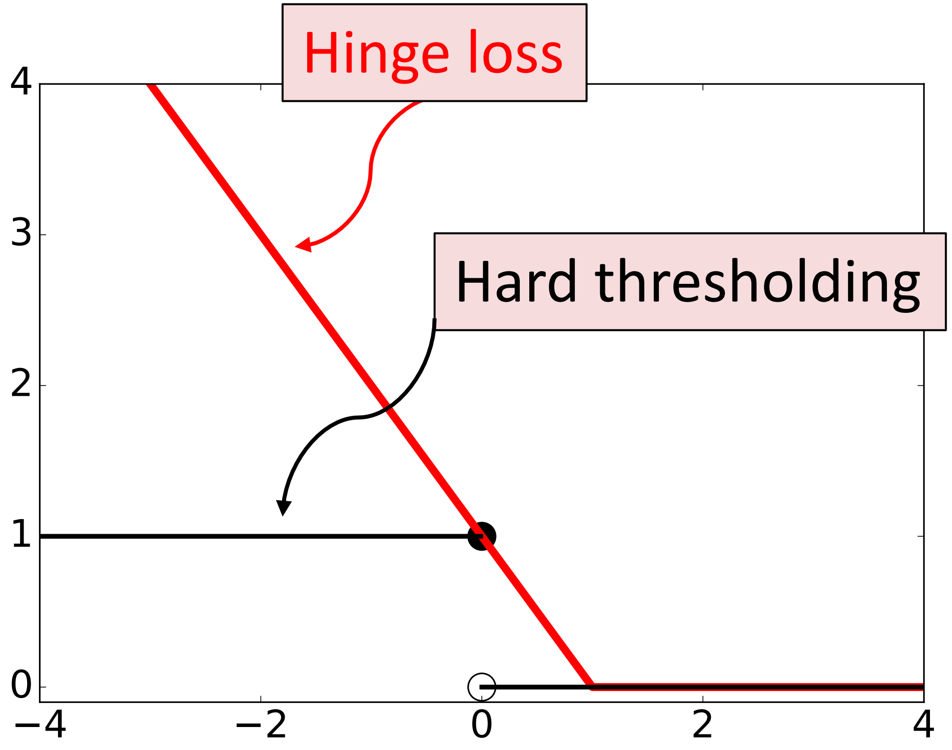
SVM: $\min_{\mathbf{w}} ||\mathbf{w}||_2^2 + \lambda \sum_j g(y_j \mathbf{w}^T \mathbf{x}_j)$.

Hinge loss: $g(z) = [1 - z]_+$.

Comparisons

SVM: $\min_{\mathbf{w}} ||\mathbf{w}||_2^2 + \lambda \sum_j g(y_j \mathbf{w}^T \mathbf{x}_j)$.

Hinge loss: $g(z) = [1 - z]_+$.

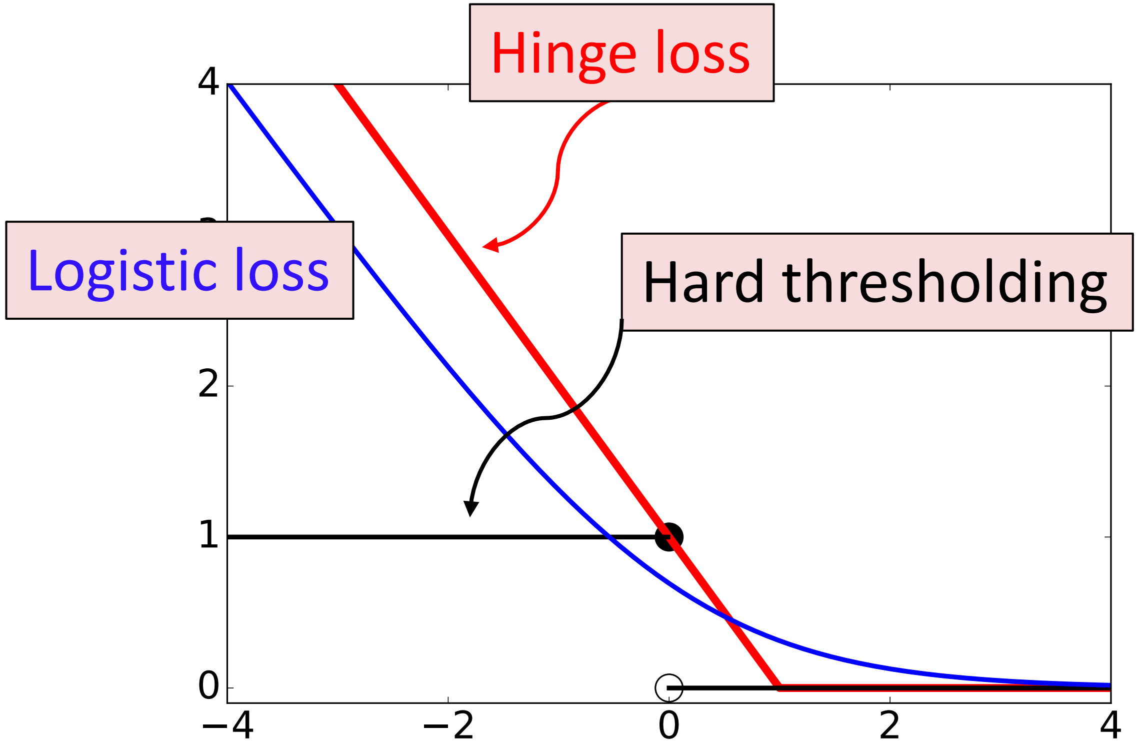


Hard thresholding: $h(\mathrm{z}) = \begin{cases} 1, & \text{if } z < 0; \\ 0, & \text{if } z \geq 0. \end{cases}$

Comparisons

SVM: $\min_{\mathbf{w}} \|\mathbf{w}\|_2^2 + \lambda \sum_j g\left(y_j \mathbf{w}^T \mathbf{x}_j\right)$.

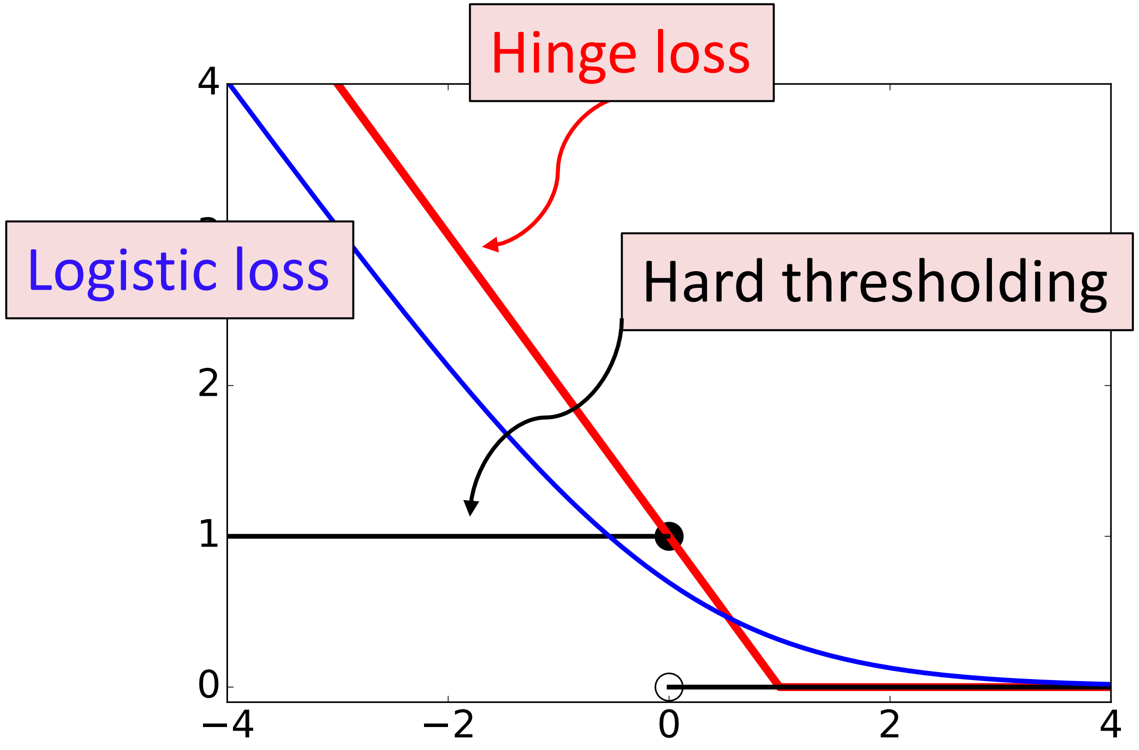
Hinge loss: $g(z) = [1 - z]_+$.



Hard thresholding: $h(z) = \begin{cases} 1, & \text{if } z < 0; \\ 0, & \text{if } z \geq 0. \end{cases}$

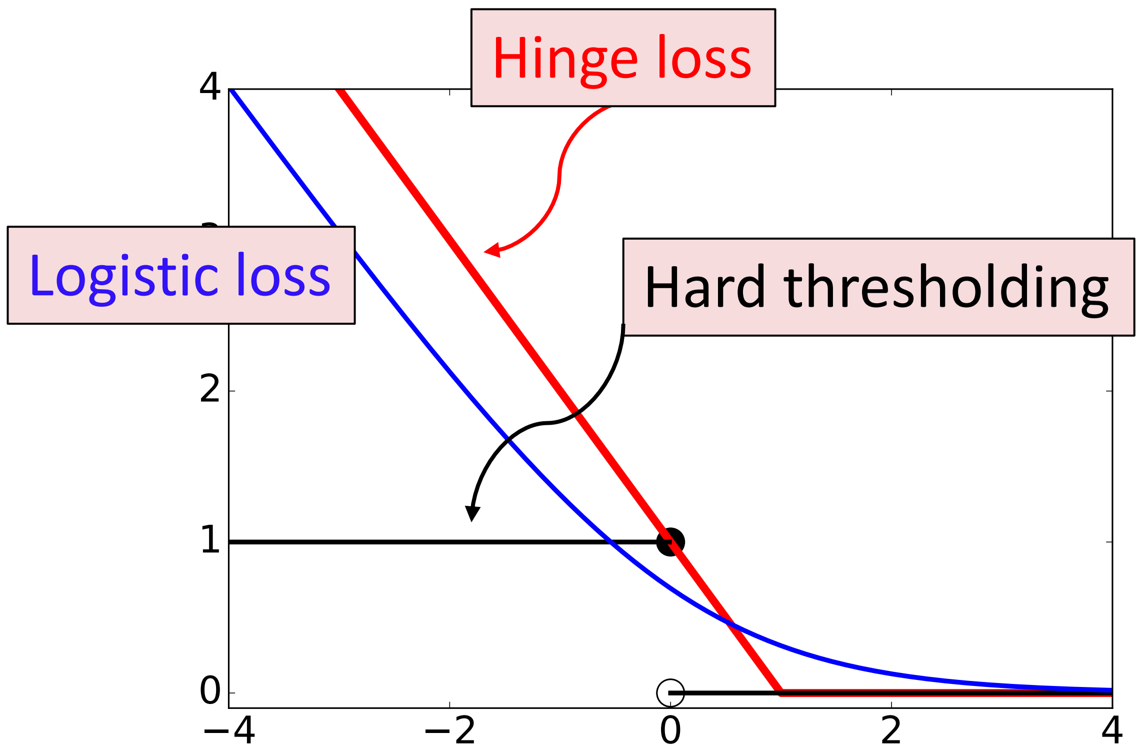
Logistic loss: $l(z) = \log(1 + e^{-z})$.

Comparisons



- Convexity
  - Hinge loss and logistic loss are convex.
  - Global optima can be efficiently found.
- Smoothness
  - Hinge loss is non-smooth.
  - Logistic loss is smooth.

Comparisons



- Convexity
  - Hinge loss and logistic loss are convex.
  - Global optima can be efficiently found.
- Smoothness
  - Hinge loss is non-smooth.
  - Logistic loss is smooth.
- Logistic regression is easier to solve than SVM.
  - GD for logistic regression has linear convergence.
  - Algorithms for SVM have sub-linear convergence.

Regularizations

The ℓ_2-Norm Regularization

Linear Regression

Input: feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ and labels $\mathbf{y} \in \mathbb{R}^{n}$.

Output: vector $\mathbf{w} \in \mathbb{R}^{d}$ such that $\mathbf{X}\mathbf{w} \approx \mathbf{y}$.

Task

Linear Regression

Input: feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ and labels $\mathbf{y} \in \mathbb{R}^{n}$.

Output: vector $\mathbf{w} \in \mathbb{R}^{d}$ such that $\mathbf{X}\mathbf{w} \approx \mathbf{y}$.

Task

- Least squares regression:

$$\min_{\mathbf{w}} \frac{1}{n} \left|\left|\mathbf{X}\mathbf{w} - \mathbf{y}\right|\right|_2^2.$$

- Ridge regression:

$$\min_{\mathbf{w}} \frac{1}{n} \left|\left|\mathbf{X}\mathbf{w} - \mathbf{y}\right|\right|_2^2 + \gamma \left|\left|\mathbf{w}\right|\right|_2^2.$$

Methods

Loss Function

Regularization

Ridge Regression: Algorithms

- **Analytical solution:** $\mathbf{w}^\star = (\mathbf{X}^T\mathbf{X} + n\gamma\mathbf{I}_d)^{-1}\mathbf{X}^T\mathbf{y}$.
  - Time complexity: $O(nd^2 + d^3)$.

Ridge Regression: Algorithms

- **Analytical solution:** $\mathbf{w}^\star = (\mathbf{X}^T\mathbf{X} + n\gamma\mathbf{I}_d)^{-1}\mathbf{X}^T\mathbf{y}$.
  - Time complexity: $O(nd^2 + d^3)$.
- Derivations:
  - The objective function is $Q(\mathbf{w}) = \frac{1}{n}\left\|\mathbf{X}\mathbf{w} - \mathbf{y}\right\|_2^2 + \gamma\left\|\mathbf{w}\right\|_2^2$.
  - The gradient is $\nabla Q(\mathbf{w}) = \frac{2}{n}\mathbf{X}^T(\mathbf{X}\mathbf{w} - \mathbf{y}) + 2\gamma\mathbf{w}$.
  - Set $\nabla Q(\mathbf{w}^\star) = 0$ leads to $\frac{2}{n}(\mathbf{X}^T\mathbf{X} + n\gamma\mathbf{I}_d)\mathbf{w}^\star = \frac{2}{n}\mathbf{X}^T\mathbf{y}$.
- Time complexity:
  - $O(nd^2)$ time for the multiplication $\mathbf{X}^T\mathbf{X}$.
  - $O(d^3)$ time for the inversion of the $d \times d$ matrix $\mathbf{X}^T\mathbf{X} + n\gamma\mathbf{I}_d$.

Ridge Regression: Algorithms

- **Conjugate gradient (CG)**
  - $O\left(\sqrt{\kappa}\log\frac{n}{\epsilon}\right)$ iterations to reach ϵ precision.
  - Hessian matrix: $\nabla^2 Q(\mathbf{w}) = \frac{2}{n}(\mathbf{X}^T\mathbf{X} + n\gamma\mathbf{I}_d)$.
  - $\kappa = \frac{\lambda_{\max}(\mathbf{X}^T\mathbf{X}) + n\gamma}{\lambda_{\min}(\mathbf{X}^T\mathbf{X}) + n\gamma}$ is the condition number of the Hessian.

Usefulness of Regularization

Question: Why do we use the ℓ_2-norm regularization?

Usefulness of Regularization

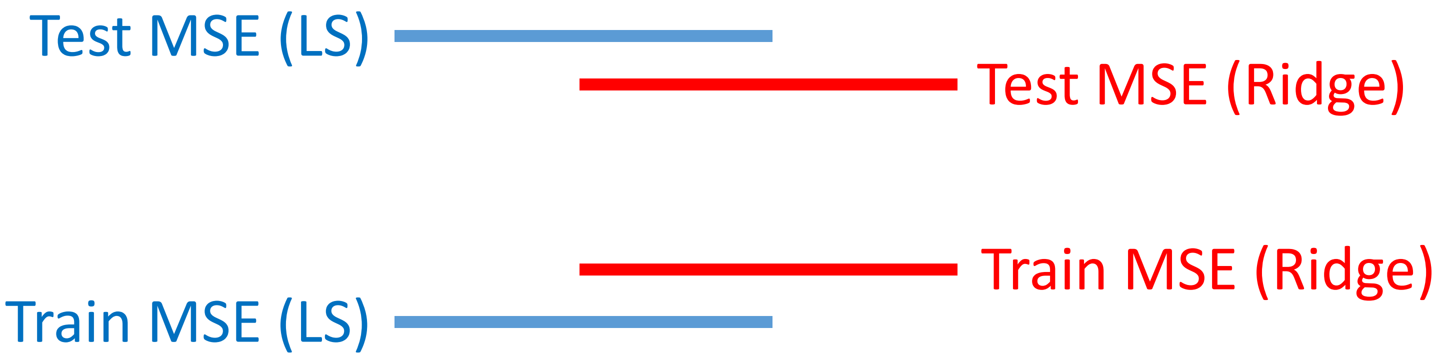
Question: Why do we use the ℓ_2-norm regularization?

- Reason 1: easier to optimize.
  - Conjugate gradient (CG) requires $O\left(\sqrt{\kappa}\log\frac{n}{\epsilon}\right)$ iterations to reach ϵ precision.
  - Least squares: $\kappa = \frac{\lambda_{\max}(\mathbf{X}^T\mathbf{X})}{\lambda_{\min}(\mathbf{X}^T\mathbf{X})}$.
  - Ridge regression: $\kappa = \frac{\lambda_{\max}(\mathbf{X}^T\mathbf{X}) + n\gamma}{\lambda_{\min}(\mathbf{X}^T\mathbf{X}) + n\gamma}$. ($\gamma \uparrow$, $\kappa \downarrow$).
  - ➡ CG converges faster as γ increases.

Usefulness of Regularization

Question: Why do we use the ℓ_2-norm regularization?

- Reason 1: easier to optimize.
- Reason 2: better generalization.
    - Least squares has better training error (due to the optimality).
    - Ridge regression makes better prediction on test set.



The ℓ_1-Norm Regularization

Motivations

$$\mathbf{x} \in \mathbb{R}^d \xrightarrow{\text{prediction}} y \in \mathbb{R}$$

Fact 1: y can be independent of some of the d feature.

Fact 2: if $d \gg n$, linear models are likely to overfit.

Motivations

$\mathbf{x} \in \mathbb{R}^d \xrightarrow{\text{prediction}} y \in \mathbb{R}$

Fact 1: y can be independent of some of the d feature.

Fact 2: if $d \gg n$, linear models are likely to overfit.

Example: Use genomic data to predict disease.

- d is huge: human have 20K protein-coding genes.
- n is small: tens or hundreds of human participants in an experiment.
- Most genes are irrelevant to a specific disease.

Motivations

$\mathbf{x} \in \mathbb{R}^d \xrightarrow{\text{prediction}} y \in \mathbb{R}$

Fact 1: y can be independent of some of the d feature.

Fact 2: if $d \gg n$, linear models are likely to overfit.

Goal 1: Select the features relevant to y.

Motivations

$$\mathbf{x} \in \mathbb{R}^d \xrightarrow{\text{prediction}} y \in \mathbb{R}$$

Fact 1: y can be independent of some of the d feature.

Fact 2: if $d \gg n$, linear models are likely to overfit.

Goal 1: Select the features relevant to y.

Goal 2: Prevent overfitting for large d, small n problems.

The ℓ_1-Norm Constraint

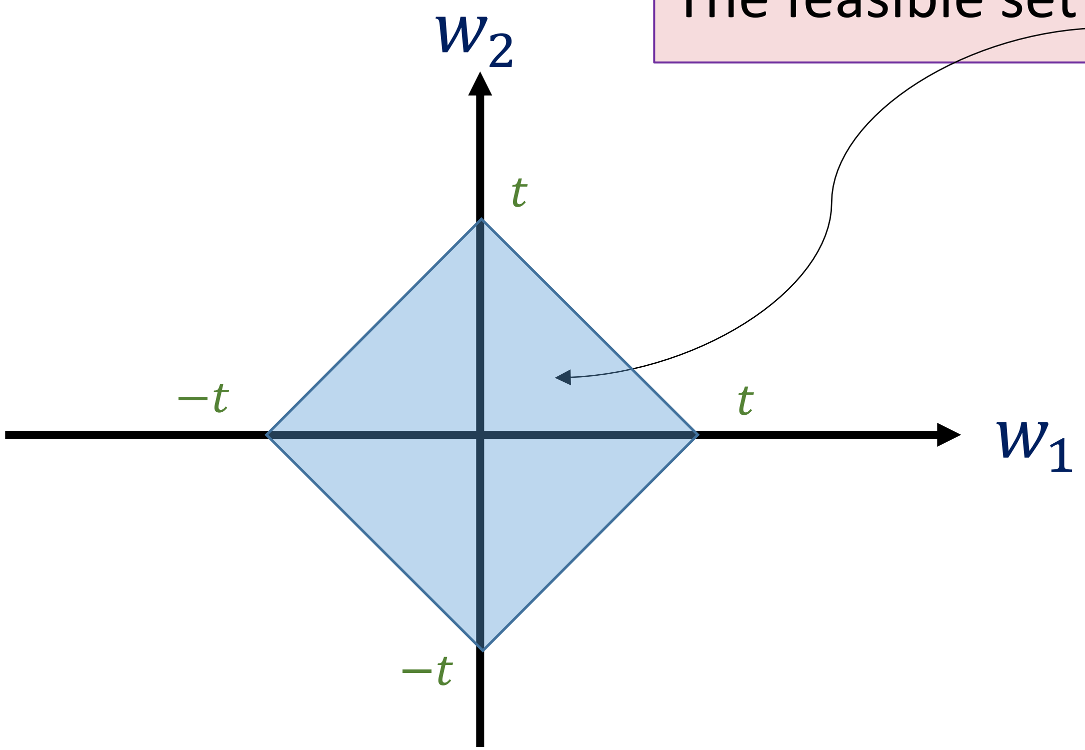
- LASSO: $\min_{\mathbf{w}} \frac{1}{2n} \left|\left|\mathbf{X}\mathbf{w} - \mathbf{y}\right|\right|_2^2;$ $\boxed{\text{s.t. } \left|\left|\mathbf{w}\right|\right|_1 \leq t.}$

The feasible set $\left\{\mathbf{w}: \left|\left|\mathbf{w}\right|\right|_1 \leq t\right\}$ is convex.

The ℓ_1-Norm Constraint

- LASSO: $\min_{\mathbf{w}} \frac{1}{2n} ||\mathbf{X}\mathbf{w} - \mathbf{y}||_2^2$; $\boxed{\text{s.t. } ||\mathbf{w}||_1 \leq t.}$

The feasible set $\{\mathbf{w}: \; ||\mathbf{w}||_1 \leq t\}$ is convex.



The ℓ_1-Norm Constraint

- LASSO: $\min_{\mathbf{w}} \frac{1}{2n} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2; \quad \text{s.t.} \ \|\mathbf{w}\|_1 \leq t.$
  - It is a convex optimization model.
  - The optimal solution $\mathbf{w}^\star$ is **sparse** (i.e., most entries are zeros).
  - Smaller t ➔ sparser $\mathbf{w}^\star$.

The ℓ_1-Norm Constraint

- LASSO: $\min_{\mathbf{w}} \; \frac{1}{2n} \big|\big|\mathbf{X}\mathbf{w} - \mathbf{y}\big|\big|_2^2 ; \qquad \text{s.t.} \; \big|\big|\mathbf{w}\big|\big|_1 \leq t.$
  - It is a convex optimization model.
  - The optimal solution $\mathbf{w}^\star$ is **sparse** (i.e., most entries are zeros).
  - Smaller t ➔ sparser $\mathbf{w}^\star$.
  - Sparsity ⟺ feature selection. Why?
    - Let $\mathbf{x}'$ be a test feature vector.
    - The prediction is $\mathbf{x}'^T \mathbf{w}^\star = w_1^\star x_1' + w_2^\star x_2' + \cdots + w_d^\star x_d'$.
    - If $w_1^\star = 0$, then the prediction is independent of x_1'.

The ℓ_1-Norm Regularization

- LASSO: $\min_{\mathbf{w}} \frac{1}{2n} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2; \quad \text{s.t.} \; \|\mathbf{w}\|_1 \leq t.$

- **Another form:** $\min_{\mathbf{w}} \underbrace{\frac{1}{2n} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2}_{\text{Loss Function}} + \underbrace{\gamma \|\mathbf{w}\|_1}_{\text{Regularization}}.$

Summary

Regularized ERM

- Regularized empirical risk minimization:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \quad \frac{1}{n}\sum_{i=1}^{n} L(\mathbf{w};\ \mathbf{x}_i, y_i) \quad + \quad R(\mathbf{w}).$$

Regularized ERM

- Regularized empirical risk minimization:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \quad \frac{1}{n}\sum_{i=1}^{n} L(\mathbf{w};\ \mathbf{x}_i, y_i) \quad + \quad R(\mathbf{w}).$$

Loss Function

- Linear regression: $L(\mathbf{w};\ \mathbf{x}_i, y_i) = \frac{1}{2}(\mathbf{w}^T\mathbf{x}_i - y_i)^2$
- Logistic regression: $L(\mathbf{w};\ \mathbf{x}_i, y_i) = \log(1 + \exp(-y_i\mathbf{w}^T\mathbf{x}_i))$
- SVM: $L(\mathbf{w};\ \mathbf{x}_i, y_i) = \max\{0,\ 1 - y_i\mathbf{w}^T\mathbf{x}_i\}$

Regularized ERM

- Regularized empirical risk minimization:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \quad \frac{1}{n}\sum_{i=1}^{n} L(\mathbf{w};\, \mathbf{x}_i, y_i) \quad + \quad R(\mathbf{w}).$$

Regularization

- ℓ_1-norm: $R(\mathbf{w}) = \gamma \|\mathbf{w}\|_1$
- ℓ_2-norm: $R(\mathbf{w}) = \gamma \|\mathbf{w}\|_2^2$
- Elastic net: $R(\mathbf{w}) = \gamma_1 \|\mathbf{w}\|_1 + \gamma_2 \|\mathbf{w}\|_2^2$

for exams wont ask proof but good to know concepts

Basics of Convex Optimization

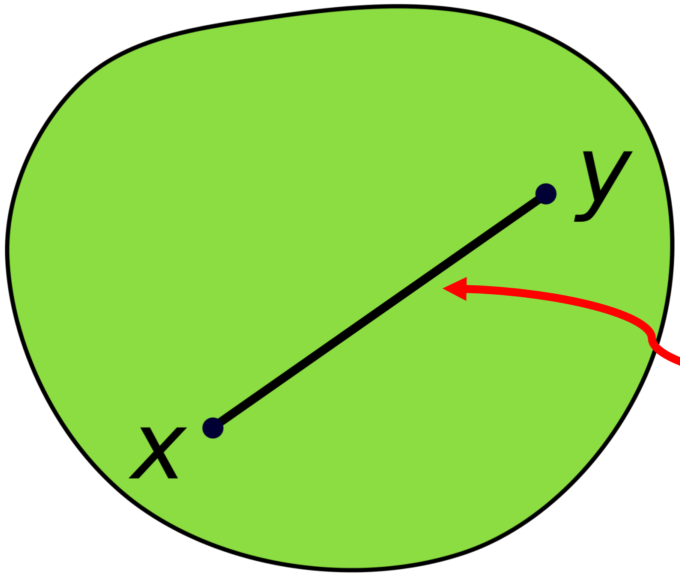
Convex Sets

Convex Set

Definition (Convex Set).

A set $\mathcal{C}$ is convex if and only if for any $\mathbf{x}, \mathbf{y} \in \mathcal{C}$ and any $\eta \in (0, 1)$, the point $\eta\mathbf{x} + (1 - \eta)\mathbf{y}$ is also in $\mathcal{C}$.

if entire line in set then its convex



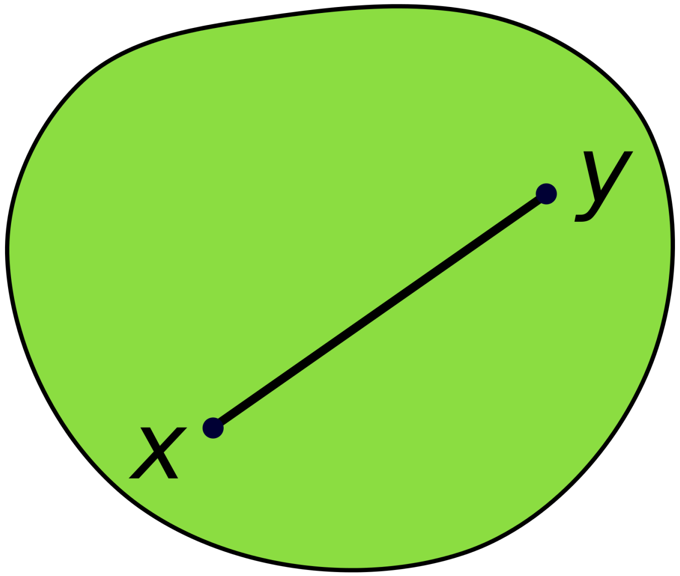
By definition, the line segment between $\mathbf{x}$ and $\mathbf{y}$ is in $\mathcal{C}$.

A convex set $\mathcal{C}$.

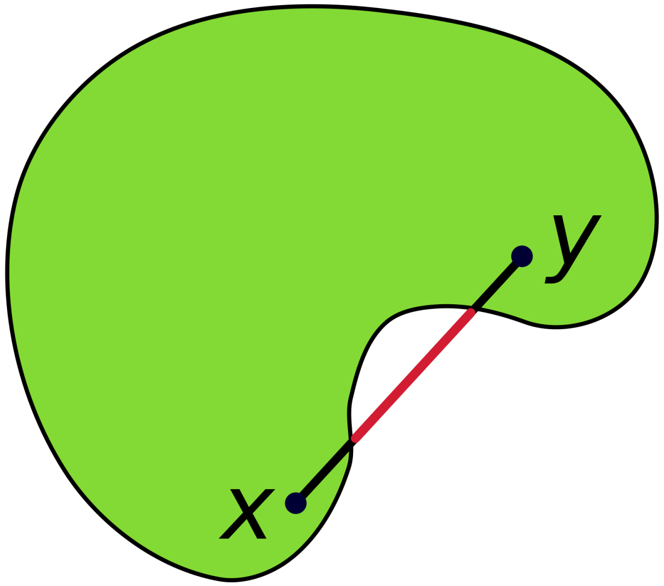
Convex Set

Definition (Convex Set).

A set $\mathcal{C}$ is convex if and only if for any $\mathbf{x}, \mathbf{y} \in \mathcal{C}$ and any $\eta \in (0, 1)$, the point $\eta\mathbf{x} + (1 - \eta)\mathbf{y}$ is also in $\mathcal{C}$.



A convex set $\mathcal{C}$.

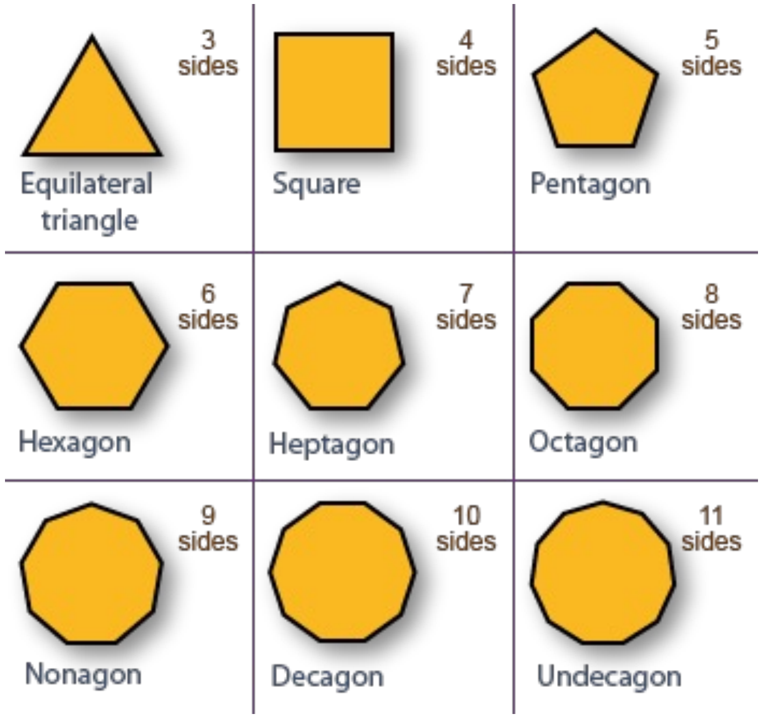


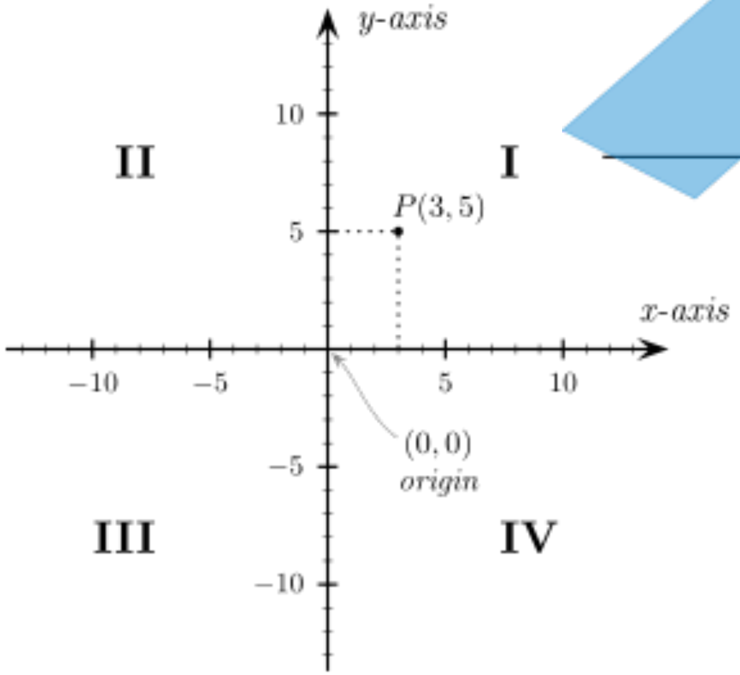
A non-convex set.

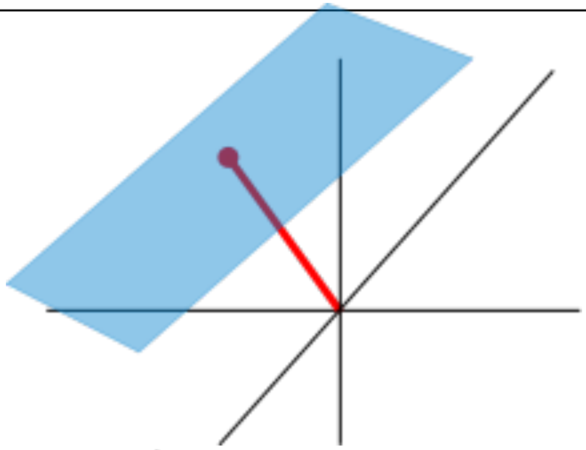
Convex Set: Examples

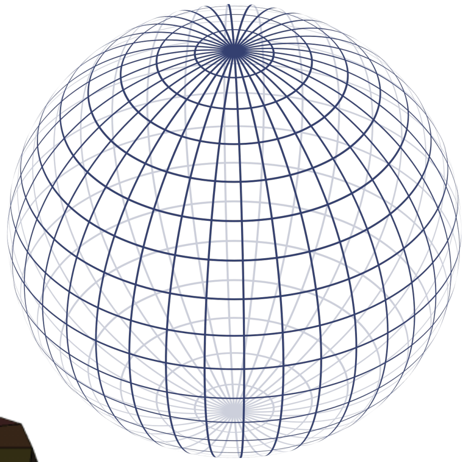
Definition (Convex Set).

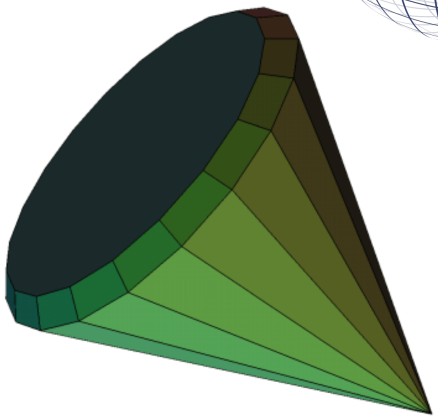
A set $\mathcal{C}$ is convex if and only if for any $\mathbf{x}, \mathbf{y} \in \mathcal{C}$ and any $\eta \in (0,1)$, the point $\eta\mathbf{x} + (1-\eta)\mathbf{y}$ is also in $\mathcal{C}$.





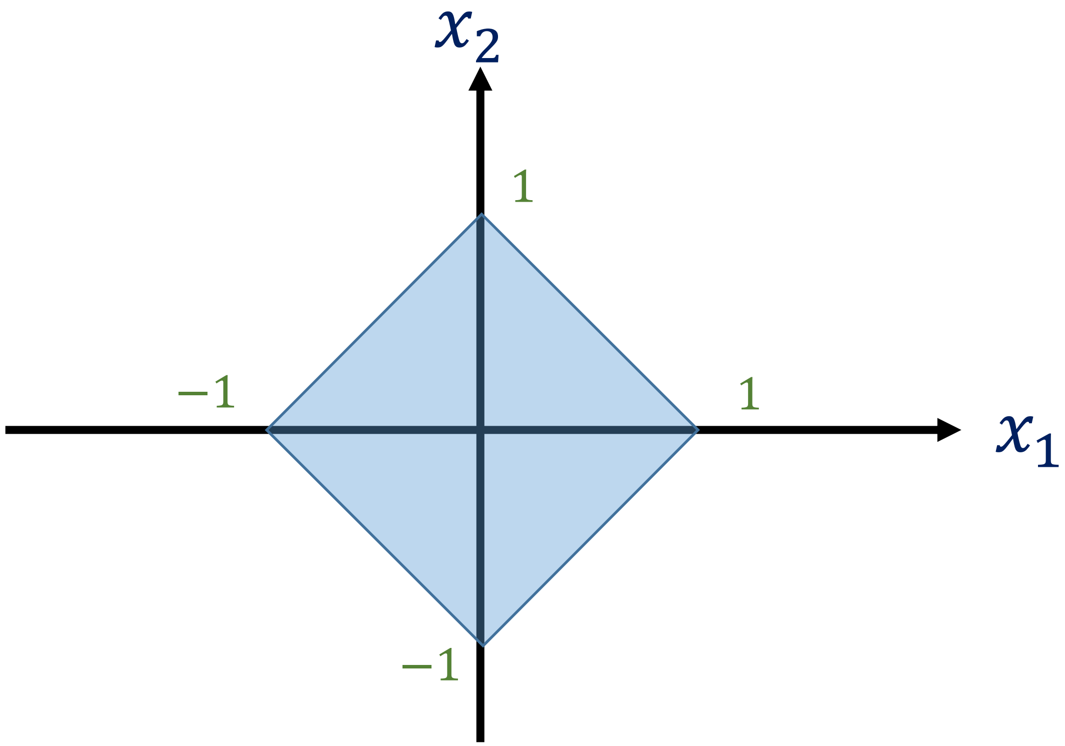






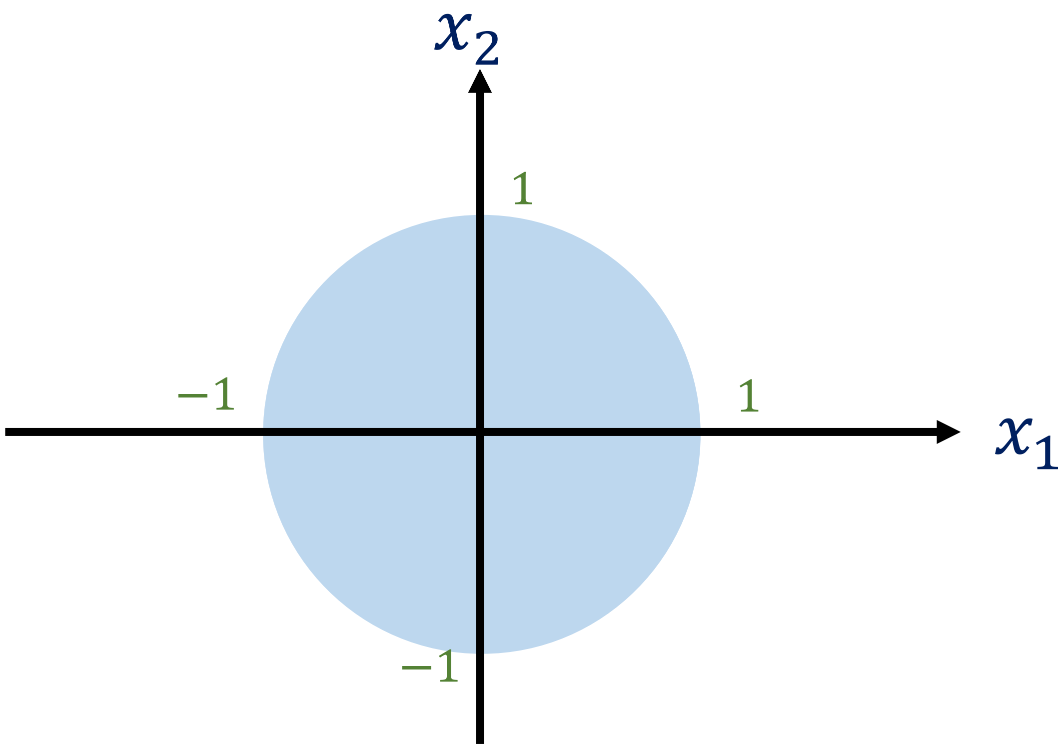
Convex Set: Examples

Example: The ℓ_1-norm ball $\left\{\mathbf{x}: \ \ ||\mathbf{x}||_1 \leq 1\right\}$.



Convex Set: Examples

Example: The ℓ_2-norm ball $\left\{\mathbf{x}: \quad ||\mathbf{x}||_2 \leq 1\right\}$.



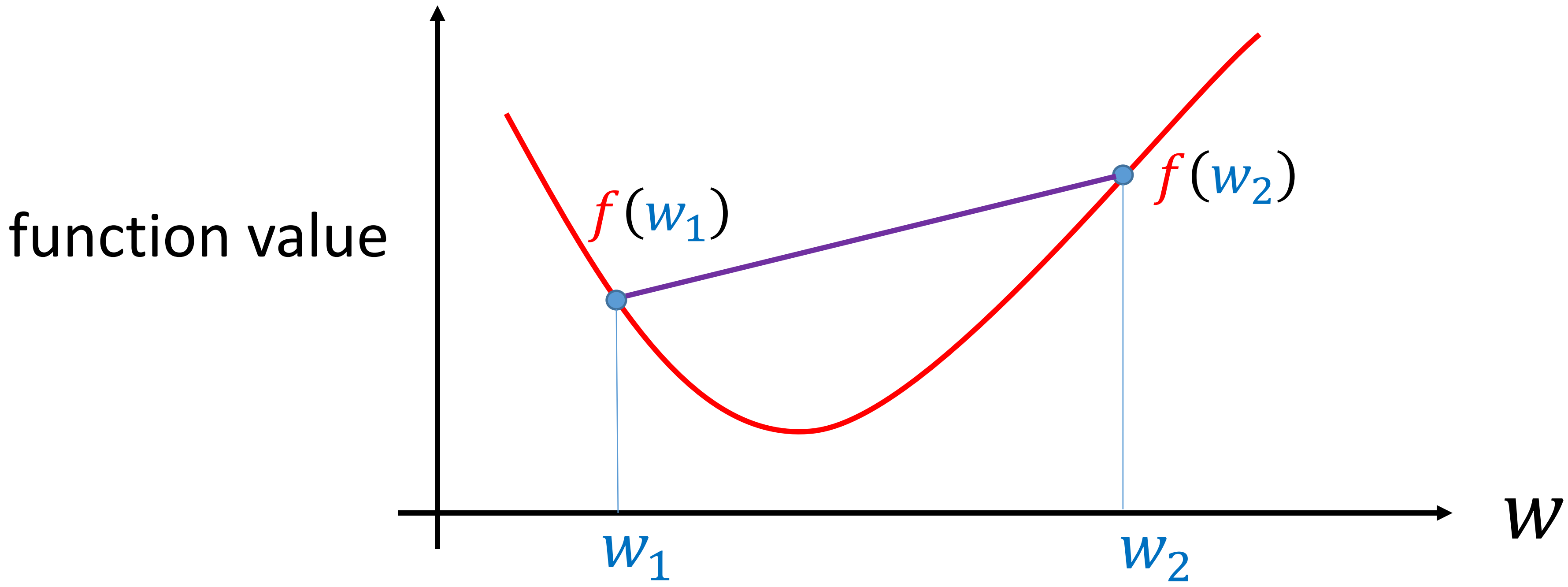
Convex Functions

Convex Function

Definition (Convex Function).

- Let $\mathcal{C}$ be a convex set and $f: \mathcal{C} \mapsto \mathbb{R}$ be a function.
- f is convex if for any $\mathbf{w}_1, \mathbf{w}_2 \in \mathcal{C}$ and any $\eta \in (0, 1)$,

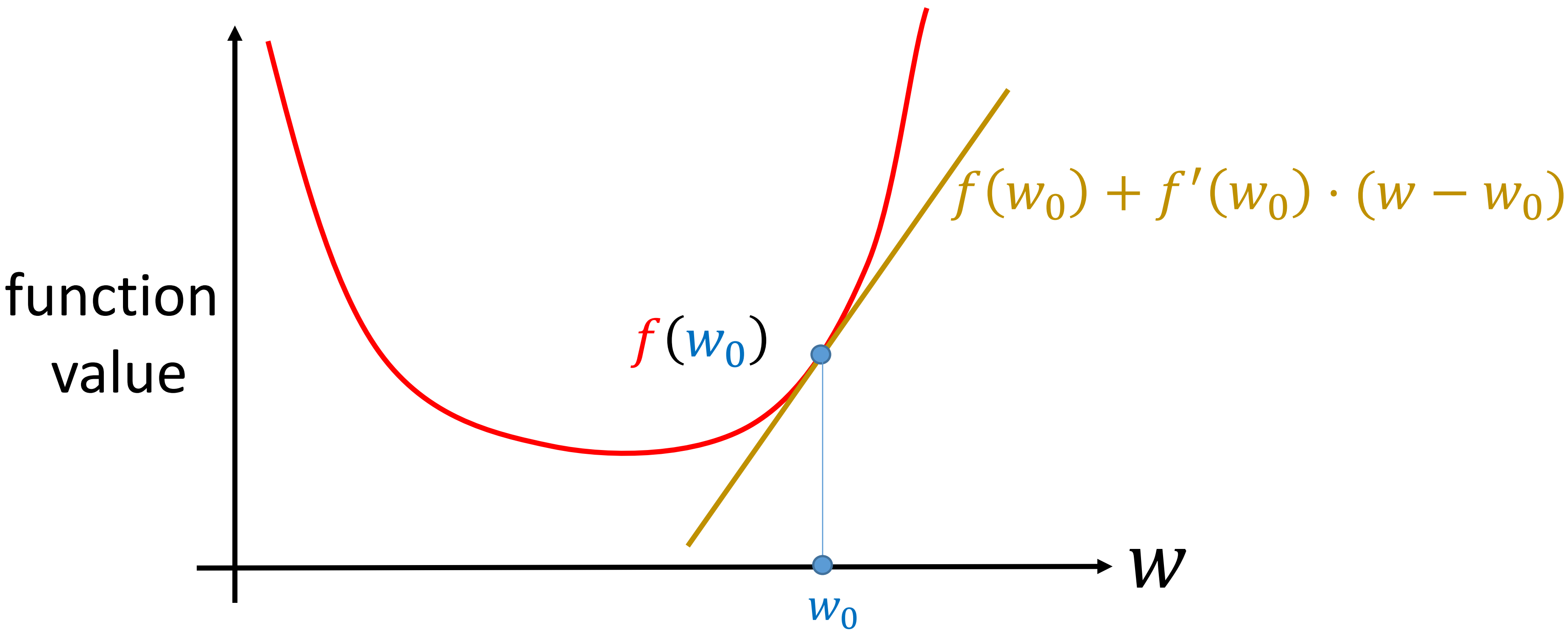
$$f(\eta \mathbf{w}_1 + (1-\eta)\mathbf{w}_2) \;\leq\; \eta f(\mathbf{w}_1) + (1-\eta) f(\mathbf{w}_2).$$



Convex Function: Properties

Properties of convex function:

1. $f(\mathbf{w}_0) + \nabla f(\mathbf{w}_0)^T(\mathbf{w} - \mathbf{w}_0) \leq f(\mathbf{w})$. (Assume f is differentiable).



Convex Function: Properties

Properties of convex function:

1. $f(\mathbf{w}_0) + \nabla f(\mathbf{w}_0)^T(\mathbf{w} - \mathbf{w}_0) \leq f(\mathbf{w})$. (Assume f is differentiable).
2. The Hessian matrix is everywhere positive semi-definite: $\nabla^2 f(\mathbf{w}) \succcurlyeq \mathbf{0}$.

- Assume f is twice differentiable.
- $\mathbf{H} \in \mathbb{R}^{d \times d}$ is positive semi-definite $\Longleftrightarrow$ for all $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{x}^T\mathbf{H}\mathbf{x} \geq 0$.

Convex Functions

Question: Are they convex functions?

- $f(w) = w^2 + w - 1$, for $w \in \mathbb{R}$.
- $f(w) = w^4$, for $w \in \mathbb{R}$.
- $f(w) = \log_e w$, for $w > 0$.
- $f(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||_2^2$, for $\mathbf{w} \in \mathbb{R}^d$.
- $f(\mathbf{w}) = \frac{1}{2}||\mathbf{X}\mathbf{w} - \mathbf{y}||_2^2$, for $\mathbf{w} \in \mathbb{R}^d$.

Convex Function: Property

Property: Combination of convex functions is convex function.

- Let $f_1, \cdots, f_k$ be convex functions.
- Then $f(\mathbf{w}) = \lambda_1 f_1(\mathbf{w}) + \cdots + \lambda_k f_k(\mathbf{w})$ is convex function for $\lambda_i \geq 0$.

Convex Function: Property

Property: Combination of convex functions is convex function.

- Let $f_1, \cdots, f_k$ be convex functions.
- Then $f(\mathbf{w}) = \lambda_1 f_1(\mathbf{w}) + \cdots + \lambda_k f_k(\mathbf{w})$ is convex function for $\lambda_i \geq 0$.

Example:

- $f_1(\mathbf{w}) = ||\mathbf{X}\mathbf{w} - \mathbf{y}||_2^2$ is convex function.
- $f_2(\mathbf{w}) = ||\mathbf{w}||_2^2$ is convex function.
- ➔ $f_1(\mathbf{w}) + \lambda f_2(\mathbf{w}) = ||\mathbf{X}\mathbf{w} - \mathbf{y}||_2^2 + \lambda||\mathbf{w}||_2^2$ is convex function.

Convex Optimization

Convex Optimization

Definition (Convex Optimization).

- Optimization: $\min_{\mathbf{w}} f(\mathbf{w})\,; \quad \text{s.t.}\ \mathbf{w} \in \mathcal{C}.$
- It is convex optimization if it has two properties:
  1. $\mathcal{C}$ (feasible set) is convex set,
  2. f (objective function) is convex function.

Convex Optimization: Examples

- Least squares regression: $\min_{\mathbf{w}} \, ||\mathbf{X}\mathbf{w} - \mathbf{y}||_2^2$.

Convex Optimization: Examples

- Least squares regression: $\min_{\mathbf{w}} \ ||\mathbf{X}\mathbf{w} - \mathbf{y}||_2^2.$
- Logistic regression: $\min_{\mathbf{w}} \sum_j \log\left(1 + \exp\left(-y_j \mathbf{w}^T \mathbf{x}_j\right)\right)$.

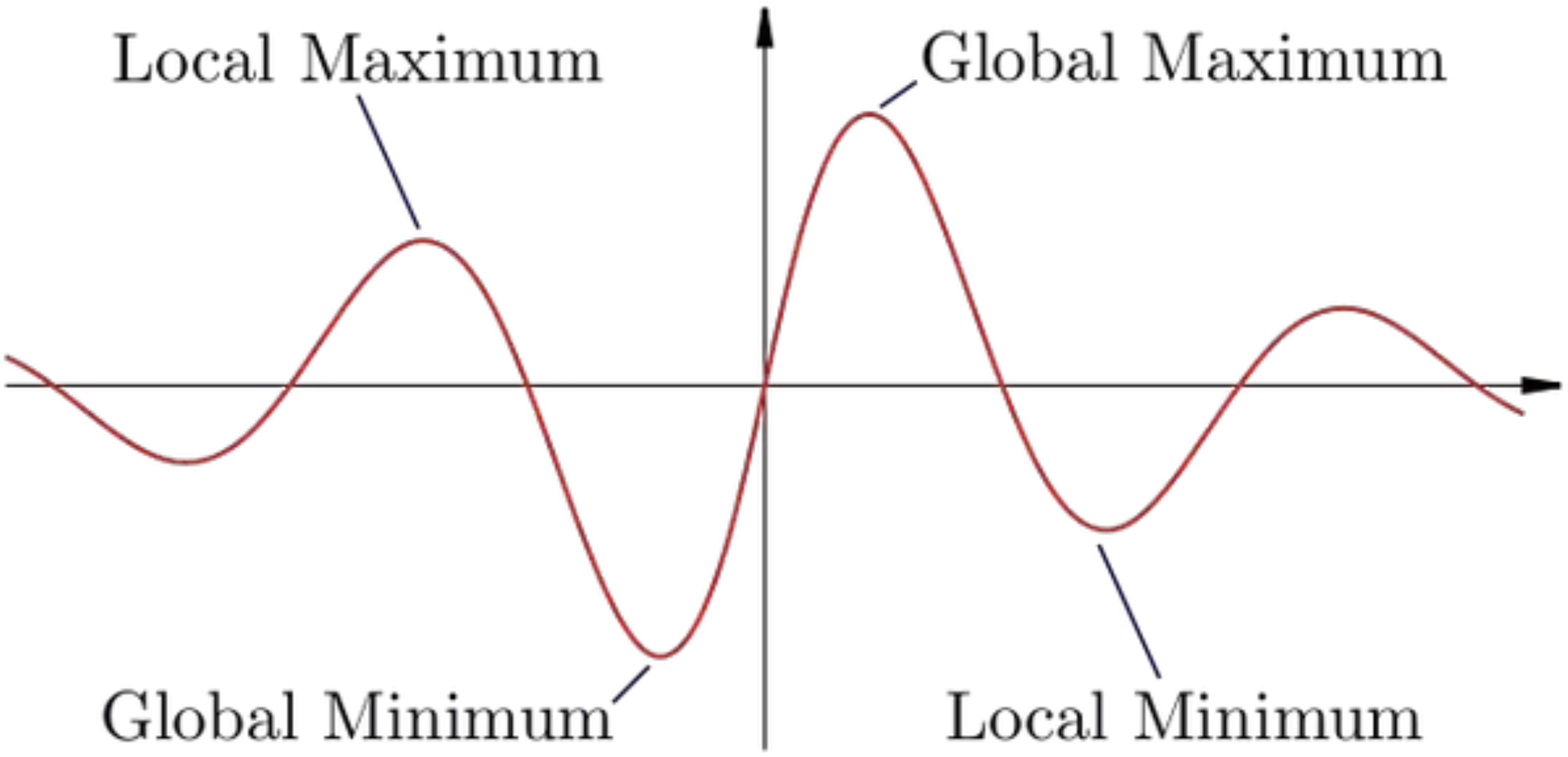
Convex Optimization: Examples

- Least squares regression: $\min_{\mathbf{w}} \; ||\mathbf{X}\mathbf{w} - \mathbf{y}||_2^2.$
- Logistic regression: $\min_{\mathbf{w}} \sum_j \log\left(1 + \exp\left(-y_j \mathbf{w}^T \mathbf{x}_j\right)\right)$.
- SVM: $\min_{\mathbf{w},b} ||\mathbf{w}||_2^2 \; + \; \lambda \sum_j \left[1 - y_j\left(\mathbf{w}^T \mathbf{x}_j + b\right)\right]_+.$

Convex Optimization: Examples

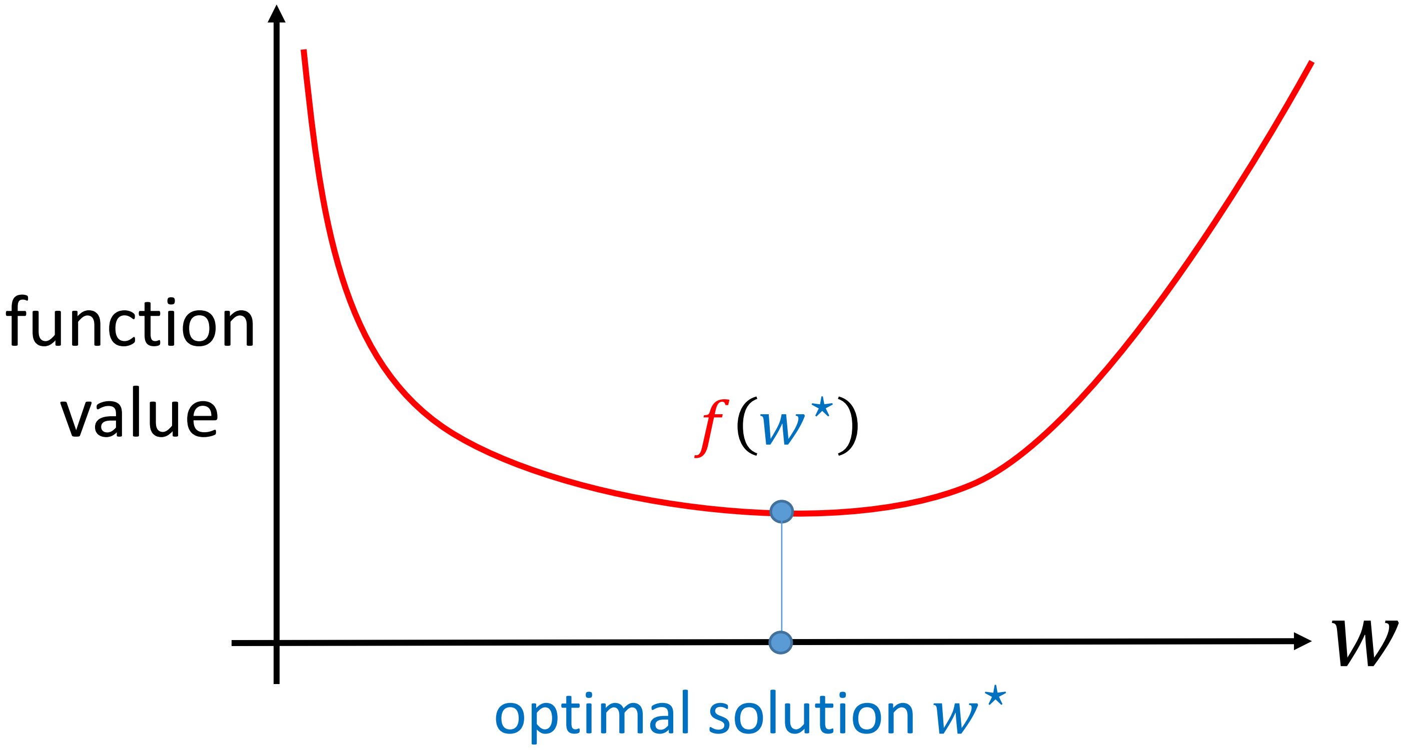
- Least squares regression: $\min_{\mathbf{w}} \ ||\mathbf{X}\mathbf{w} - \mathbf{y}||_2^2$.
- Logistic regression: $\min_{\mathbf{w}} \sum_j \log\left(1 + \exp\left(-y_j \mathbf{w}^T \mathbf{x}_j\right)\right)$.
- SVM: $\min_{\mathbf{w},b} ||\mathbf{w}||_2^2 \ + \ \lambda \sum_j \left[1 - y_j\left(\mathbf{w}^T \mathbf{x}_j + b\right)\right]_+$.
- LASSO: $\min_{\mathbf{w}} \ ||\mathbf{X}\mathbf{w} - \mathbf{y}||_2^2; \quad s.t.\ ||\mathbf{w}||_1 \leq t$.

Local and Global Optima



Convex Optimization: Properties

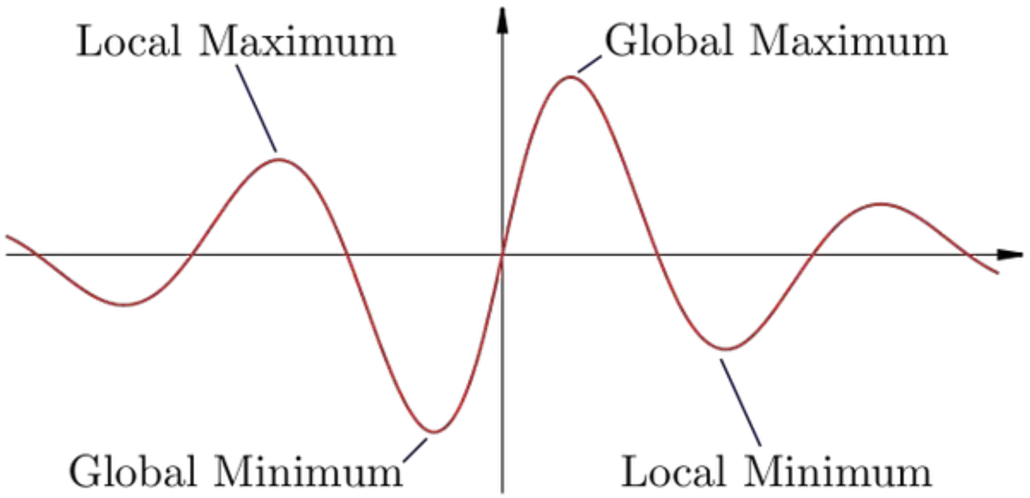
Property: For convex optimization, every local minimum is global minimum.



Optimization: Properties

First-order optimality condition (necessary condition):

- Consider the unconstrained optimization: $\min_{\mathbf{w}} f(\mathbf{w})$.
- If $\mathbf{w}^\star$ is local minimum, then the gradient $\frac{\partial f(\mathbf{w})}{\partial \mathbf{w}}$ at $\mathbf{w}^\star$ is zero.



Convex Optimization: Properties

First-order optimality condition (necessary condition):

- Consider the unconstrained optimization: $\min_{\mathbf{w}} f(\mathbf{w})$.
- If $\mathbf{w}^\star$ is local minimum, then the gradient $\frac{\partial f(\mathbf{w})}{\partial \mathbf{w}}$ at $\mathbf{w}^\star$ is zero.

Property of convex optimization (sufficient condition):

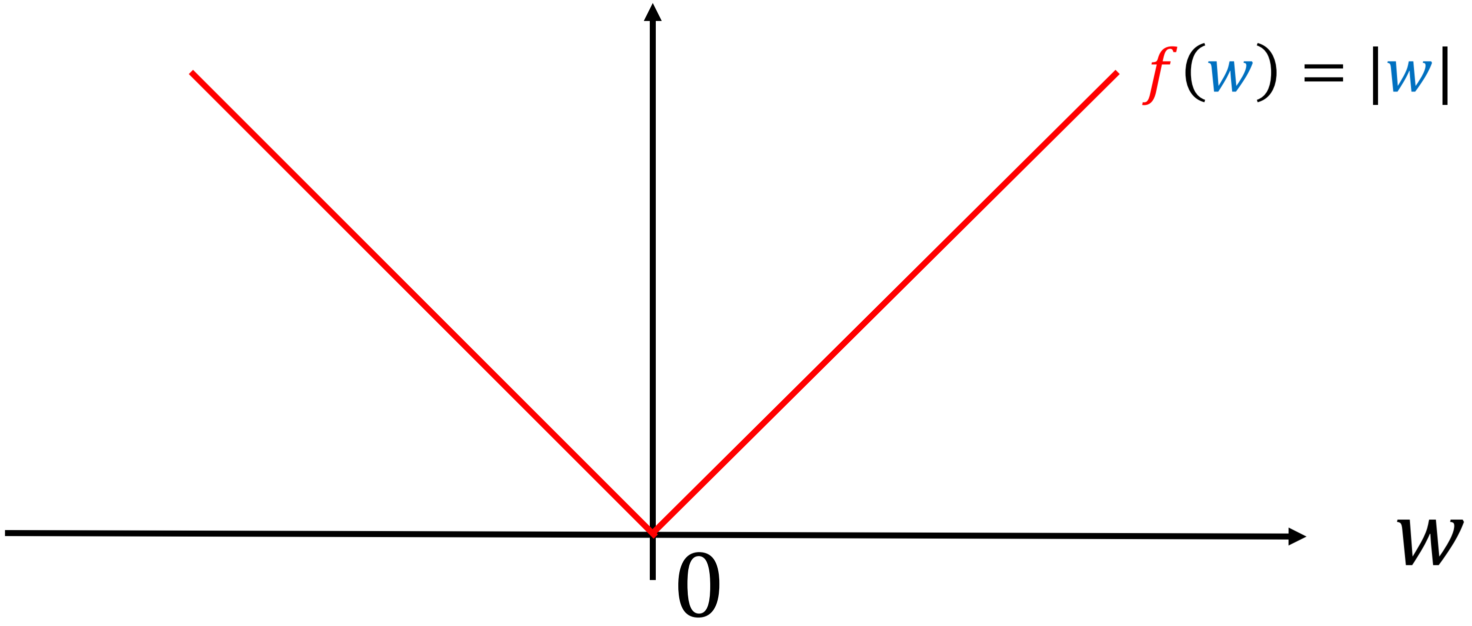
- Let $\min_{\mathbf{w}} f(\mathbf{w})$ be convex optimization.
- If $\frac{\partial f(\mathbf{w})}{\partial \mathbf{w}}$ at $\mathbf{w}^\star$ is zero, then $\mathbf{w}^\star$ is global minimum.

Subgradient and Subdifferential

Non-Differentiable Functions

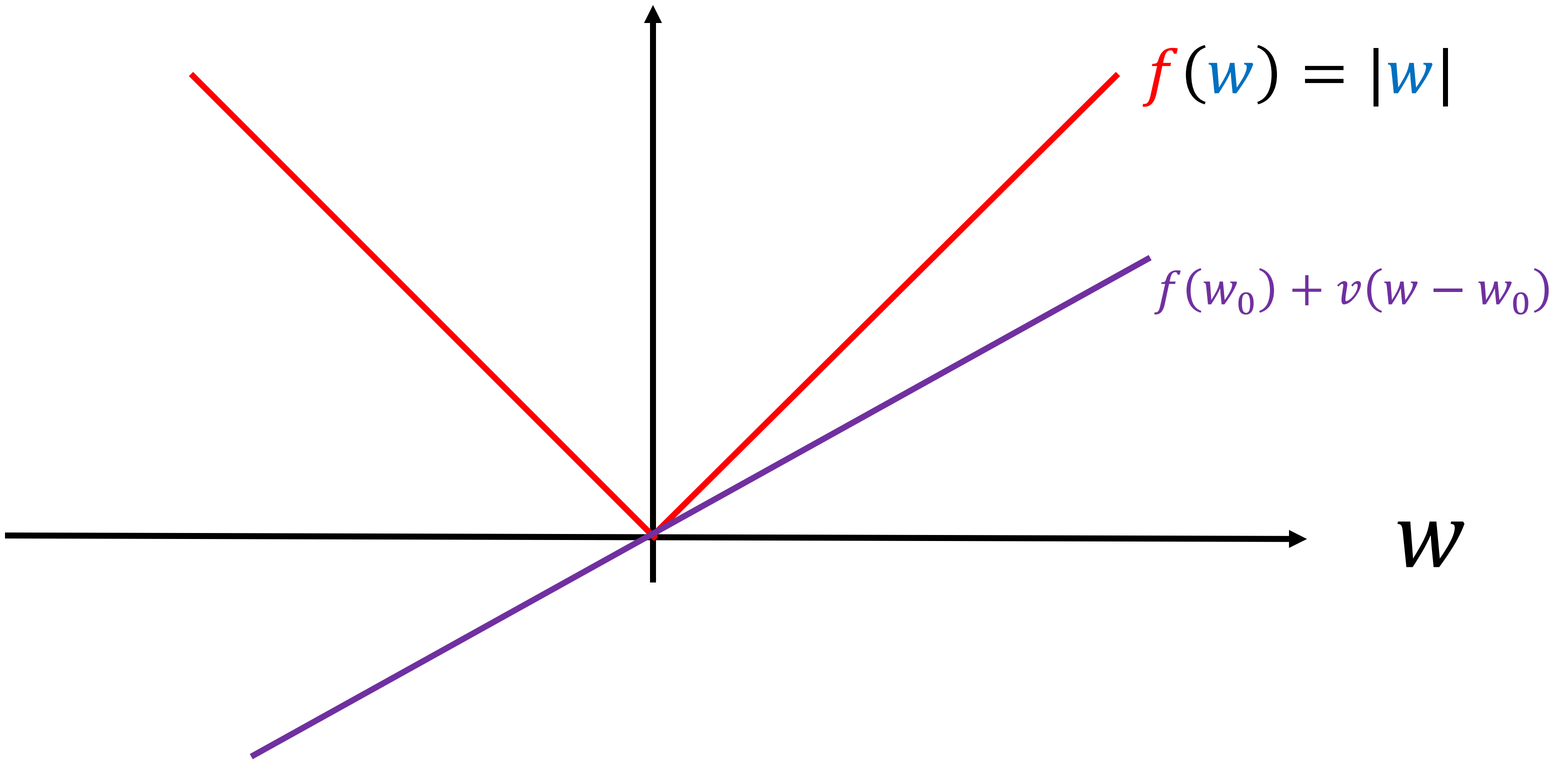
- Example of non-differentiable functions: $f(w) = |w|$

$$\frac{\partial f}{\partial w} = \begin{cases} +1, & \text{if } w > 0; \\ \text{undefined}, & \text{if } w = 0; \\ -1, & \text{if } w < 0. \end{cases}$$



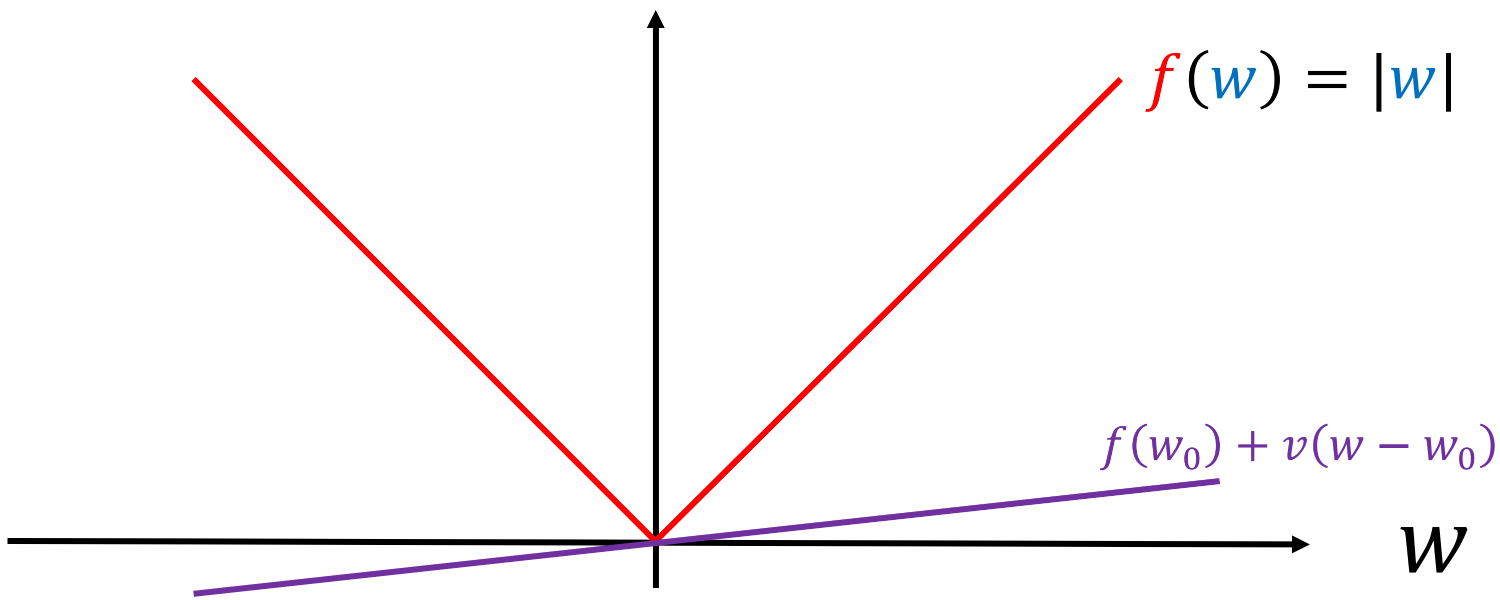
Subgradient of Convex Function

Definition (Subgradient). A vector $\mathbf{v}$ is called a subgradient of f at $\mathbf{w}_0$ if for any $\mathbf{w}$, $f(\mathbf{w}) \geq f(\mathbf{w}_0) + \mathbf{v}^T(\mathbf{w} - \mathbf{w}_0)$.



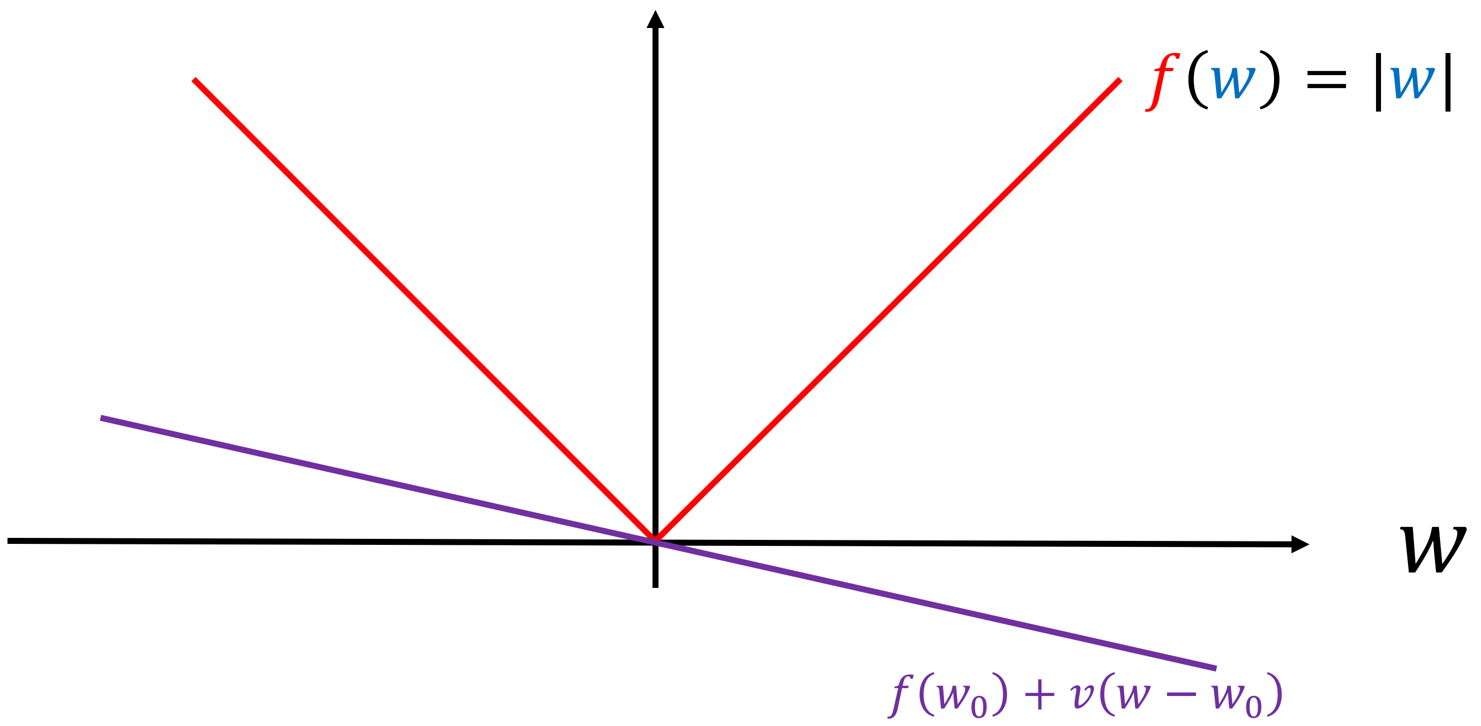
Subgradient of Convex Function

Definition (Subgradient). A vector $\mathbf{v}$ is called a subgradient of f at $\mathbf{w}_0$ if for any $\mathbf{w}$, $f(\mathbf{w}) \geq f(\mathbf{w}_0) + \mathbf{v}^T(\mathbf{w} - \mathbf{w}_0)$.



Subgradient of Convex Function

Definition (Subgradient). A vector $\mathbf{v}$ is called a subgradient of f at $\mathbf{w}_0$ if for any $\mathbf{w}$, $f(\mathbf{w}) \geq f(\mathbf{w}_0) + \mathbf{v}^T(\mathbf{w} - \mathbf{w}_0)$.



Subdifferential of Convex Function

Definition (Subgradient). A vector $\mathbf{v}$ is called a subgradient of f at $\mathbf{w}_0$ if for any $\mathbf{w}$, $f(\mathbf{w}) \geq f(\mathbf{w}_0) + \mathbf{v}^T(\mathbf{w} - \mathbf{w}_0)$.

Definition (Subdifferential). The set containing all the subgradients of f at $\mathbf{w}_0$ is called the subdifferential. Denote the set by $\partial f(\mathbf{w}_0)$.

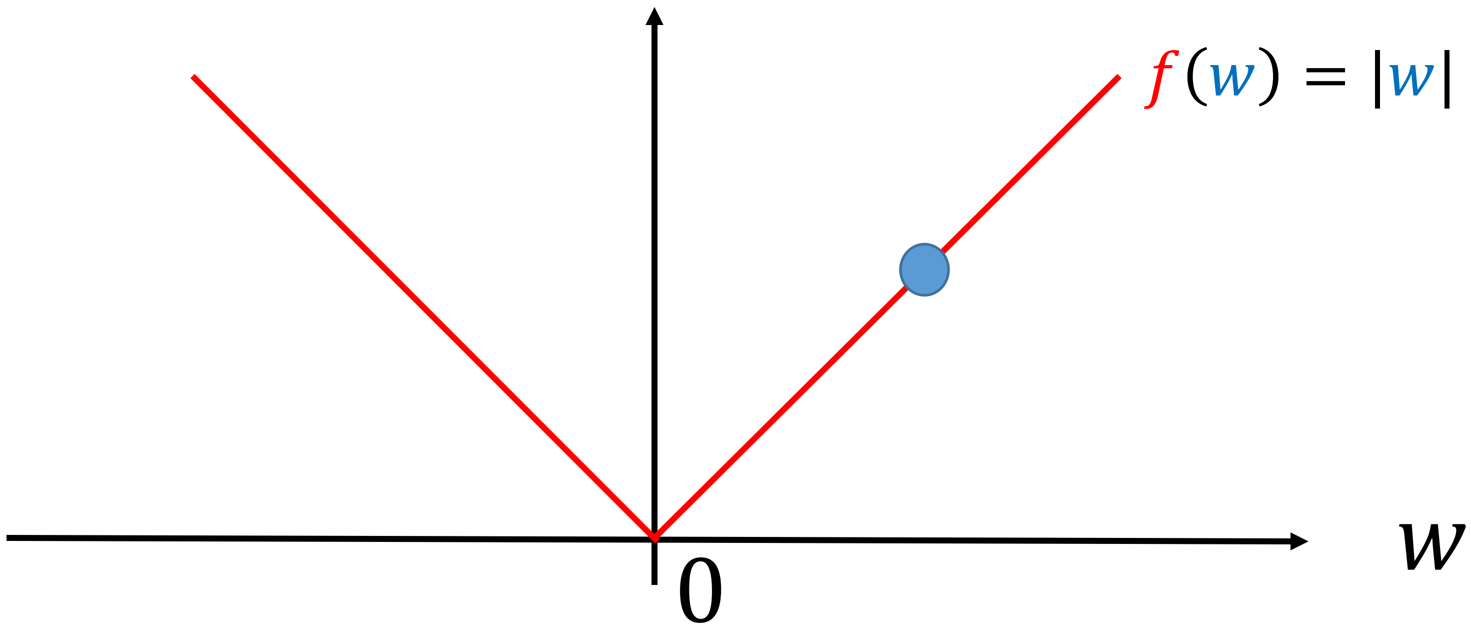
Subdifferential of Convex Function

Definition (Subgradient). A vector $\mathbf{v}$ is called a subgradient of f at $\mathbf{w}_0$ if for any $\mathbf{w}$, $f(\mathbf{w}) \geq f(\mathbf{w}_0) + \mathbf{v}^T(\mathbf{w} - \mathbf{w}_0)$.

Definition (Subdifferential). The set containing all the subgradients of f at $\mathbf{w}_0$ is called the subdifferential. Denote the set by $\partial f(\mathbf{w}_0)$.

Example: $f(w) = |w|$

- $\partial f(3) = \{1\}$.



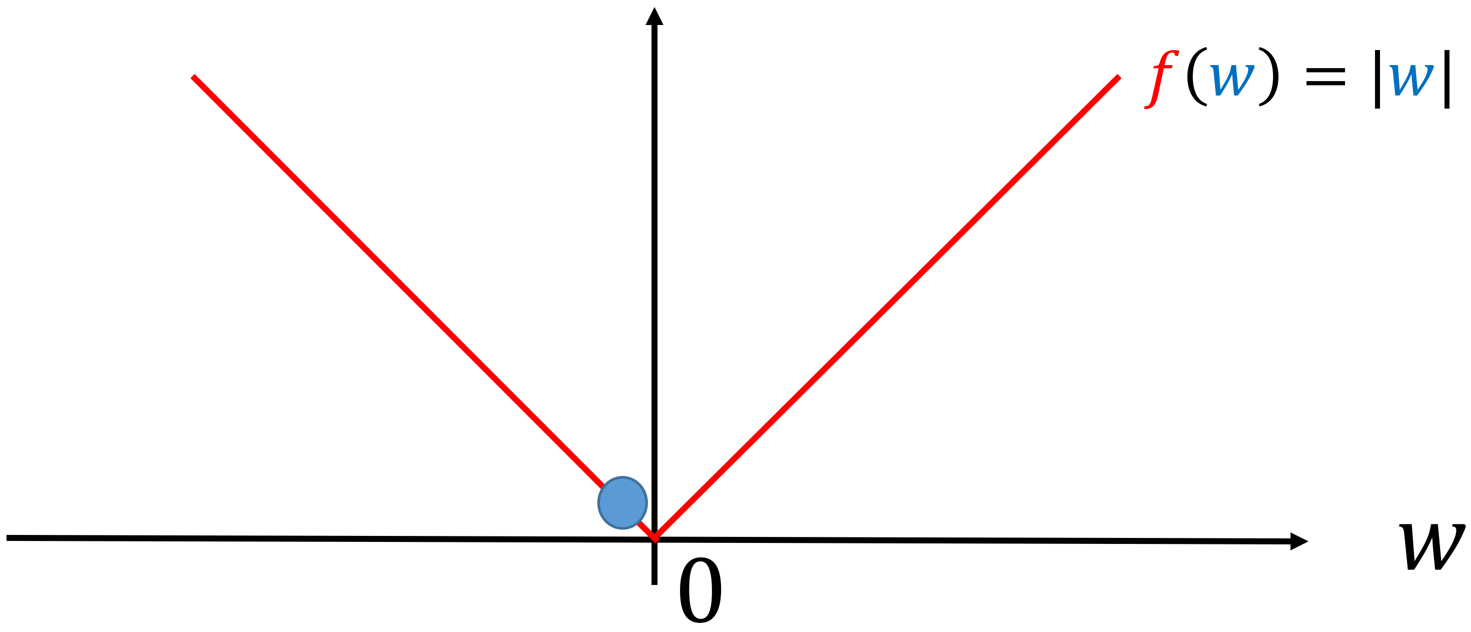
Subdifferential of Convex Function

Definition (Subgradient). A vector $\mathbf{v}$ is called a subgradient of f at $\mathbf{w}_0$ if for any $\mathbf{w}$, $f(\mathbf{w}) \geq f(\mathbf{w}_0) + \mathbf{v}^T(\mathbf{w} - \mathbf{w}_0)$.

Definition (Subdifferential). The set containing all the subgradients of f at $\mathbf{w}_0$ is called the subdifferential. Denote the set by $\partial f(\mathbf{w}_0)$.

Example: $f(w) = |w|$

- $\partial f(3) = \{1\}$.
- $\partial f(-0.1) = \{-1\}$.



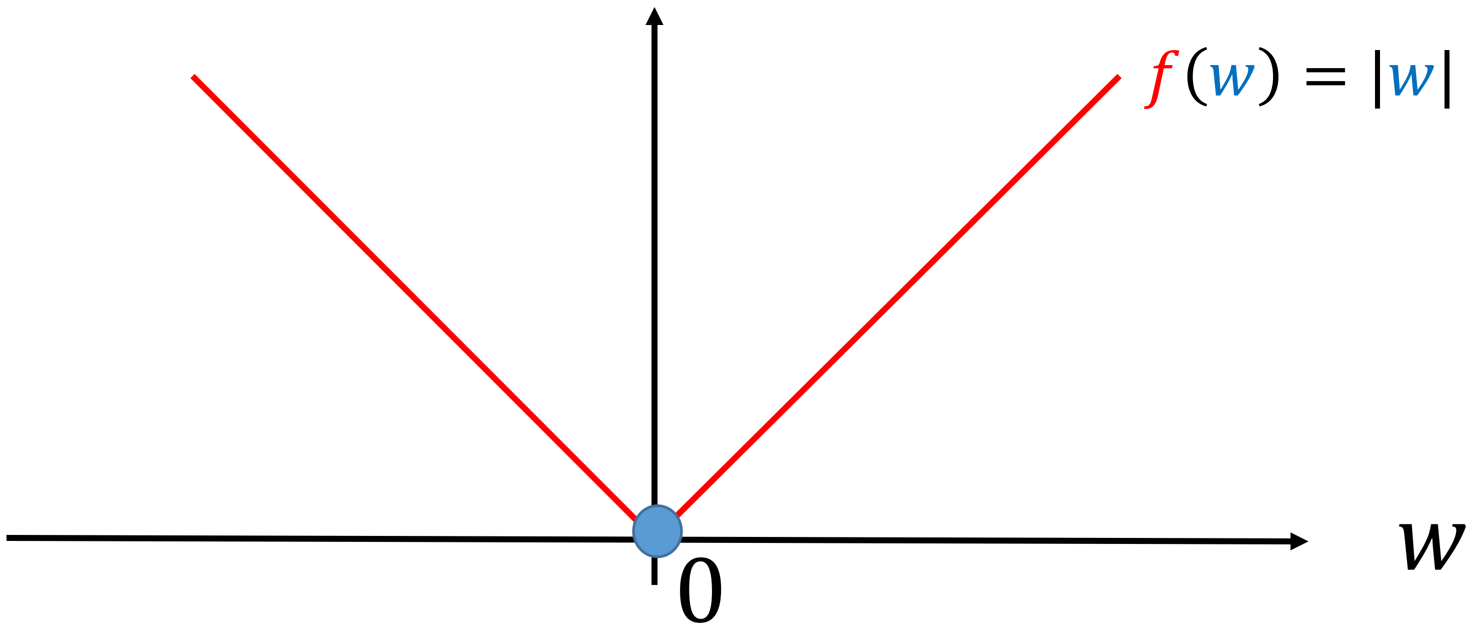
Subdifferential of Convex Function

Definition (Subgradient). A vector $\mathbf{v}$ is called a subgradient of f at $\mathbf{w}_0$ if for any $\mathbf{w}$, $f(\mathbf{w}) \geq f(\mathbf{w}_0) + \mathbf{v}^T(\mathbf{w} - \mathbf{w}_0)$.

Definition (Subdifferential). The set containing all the subgradients of f at $\mathbf{w}_0$ is called the subdifferential. Denote the set by $\partial f(\mathbf{w}_0)$.

Example: $f(w) = |w|$

- $\partial f(3) = \{1\}$.
- $\partial f(-0.1) = \{-1\}$.
- $\partial f(0) = [-1, 1]$.



A Property of Convex Optimization

Let f be a convex function.

Property: $\mathbf{w}^\star = \min_{\mathbf{w}} f(\mathbf{w}) \Longleftrightarrow 0 \in \partial f(\mathbf{w}^\star)$.

Example: $\min_{w} \{f(w) = |w + 5|\}$

- $\partial f(-5) = [-1, 1]$.
- Obviously $0 \in \partial f(-5)$.
- $w^\star = -5$ minimizes f .