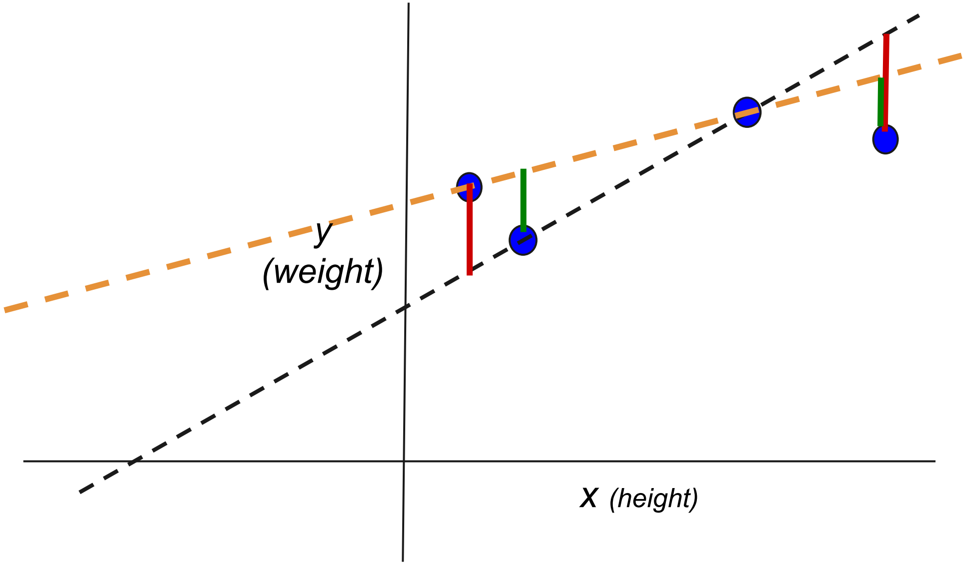
CIS 678 Machine Learning

ML Models: SVM, Kernel Methods

Regression (LR)



Model

$$\hat{y} = \beta_0 + \beta_1 x$$

$$\Theta = \{\beta_0, \beta_1\}$$

Fitting Error

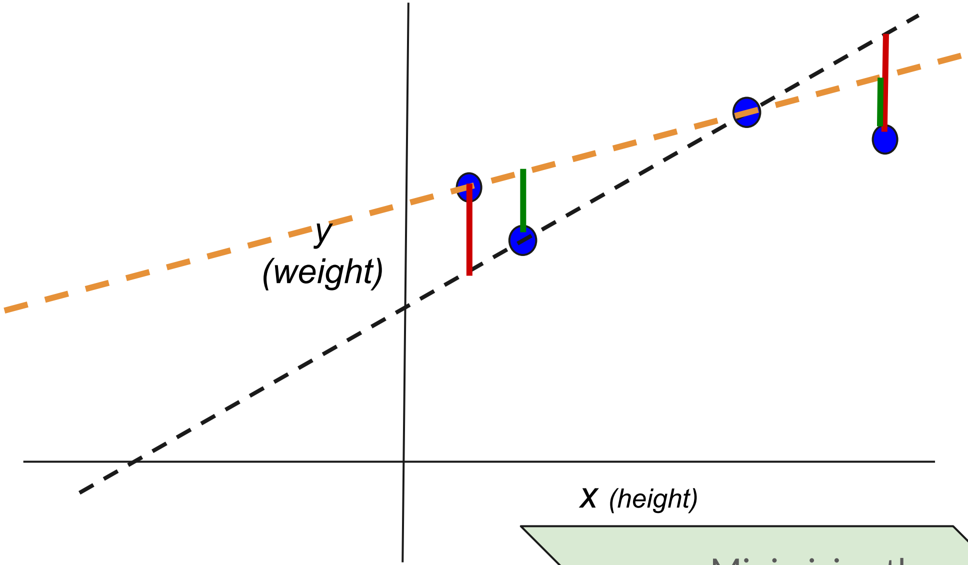
$$\epsilon = |\hat{y} - y|$$

Optimization function

$$E_\Theta = \frac{1}{2}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

$$\Theta^* = \mathrm{argmin}_\Theta\, E\{(x_i, y_i)\}_{i=1,\cdots,N}$$

Regression (LR)



Minimizing the Quadratic Loss; right?

Model

$$\hat{y} = \beta_0 + \beta_1 x$$

$$\Theta = \{\beta_0, \beta_1\}$$

Fitting Error

$$\epsilon = |\hat{y} - y|$$

Optimization function

$$E_\Theta = \frac{1}{2} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2$$

$$\Theta^* = \mathrm{argmin}_\Theta \, E\{(x_i, y_i)\}_{i=1,\cdots,N}$$

Loss functions

Regression

- Quadratic (L2) loss
  - Mean Squared Error (MSE)
- Absolute (L1) loss
  - Mean Absolute Error (MAE
- MAPE

$$\boxed{MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

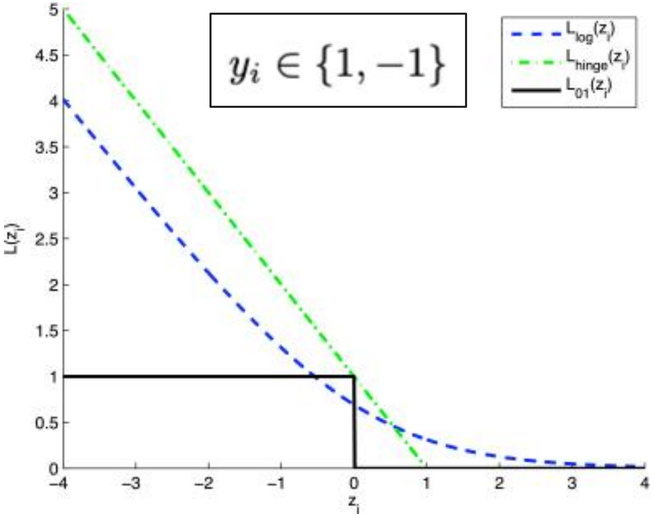
$$\boxed{MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|}$$

Loss functions

Classification

- Misclassification rate (0-1 loss)
- Log loss
- Hinge loss
- Cross entropy loss

Loss function



$$y_i \in \{1, -1\}$$

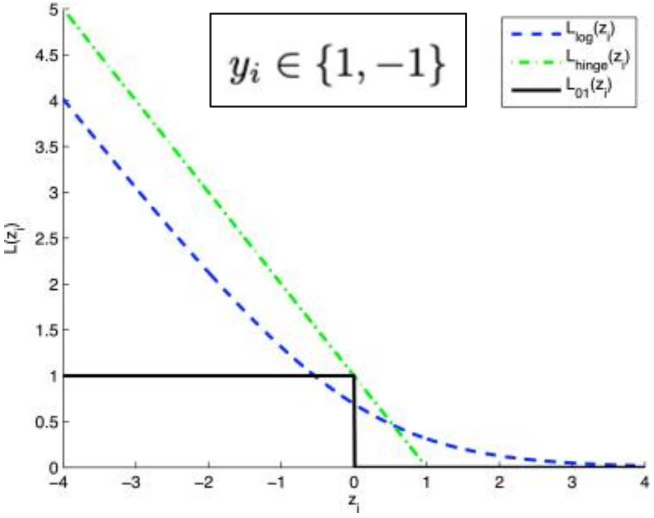
Three widely used loss functions as a function of their input (z_i): the log logistic loss, the hinge loss, 01 loss

Classification

- **Misclassification rate (0-1 loss)**
- Log loss
- Hinge loss
- Cross entropy loss

$$\boxed{L_{01}(z_i) = \mathbb{I}[z_i \leq 0],}$$

Loss function



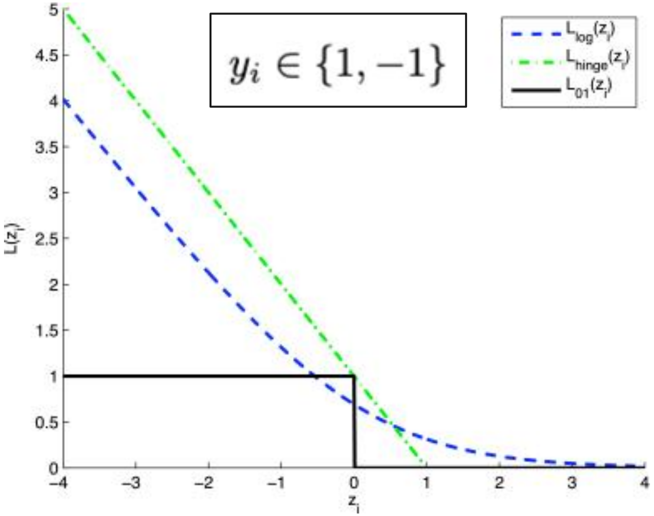
Three widely used loss functions as a function of their input (z_i): the log logistic loss, the hinge loss, 01 loss

Classification

- Misclassification rate (0-1 loss)
- **Log loss**
- Hinge loss
- Cross entropy loss

$$\boxed{L_{log}(z_i) = \log[1 + \exp(-z_i)]}$$

Loss function



Three widely used loss functions as a function of their input (z_i): the log logistic loss, the hinge loss, 01 loss

Classification

- Misclassification rate (0-1 loss)
- Log loss
- Hinge loss
- Cross entropy loss

$$L_{hinge}(z_i) = \max(0, 1 - z_i)$$

Loss function

$$\boxed{y_i \in \{0, 1\}}$$

- Encourages the model to output higher probabilities for the positive class and lower probabilities for the negative class.

Classification

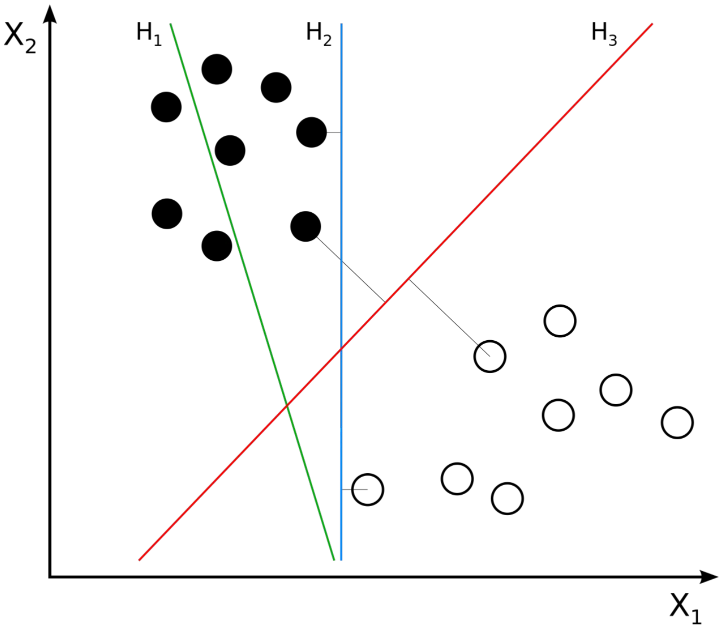
- Misclassification rate (0-1 loss)
- Log loss
- Hinge loss
- **Cross entropy loss**

$$\boxed{L = \frac{1}{N}\sum_{i=1}^{N}(y_i \log(p_i) + (1 - y_i)\log(1 - p_i))}$$

Support Vector Machines

- Maximum margin models

Motivation



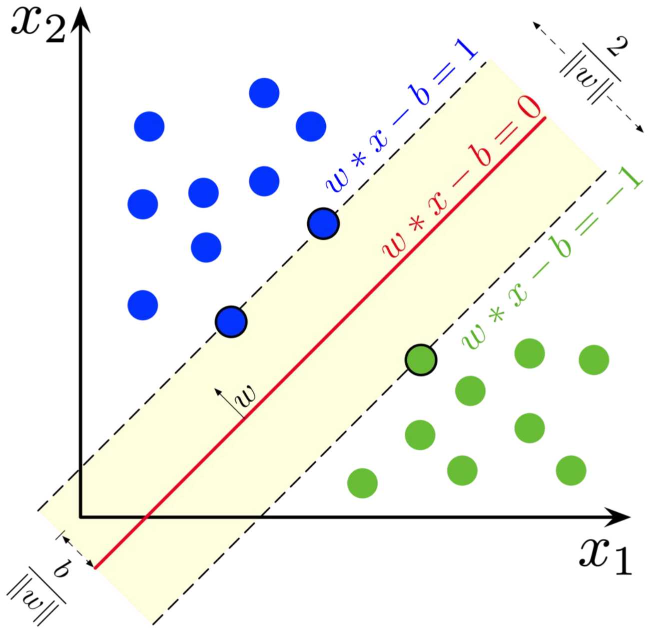
H_1 does not separate the classes. H_2 does, but only with a small margin. H_3 separates them with the maximal margin. (Wiki)

Linear SVM

We are given a training dataset of n points of the form

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n),$$

$$\boxed{y_i \in \{1, -1\}}$$



Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors.(Wiki)

Linear SVM

We are given a training dataset of n points of the form

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n),$$

$$\boxed{y_i \in \{1, -1\}}$$

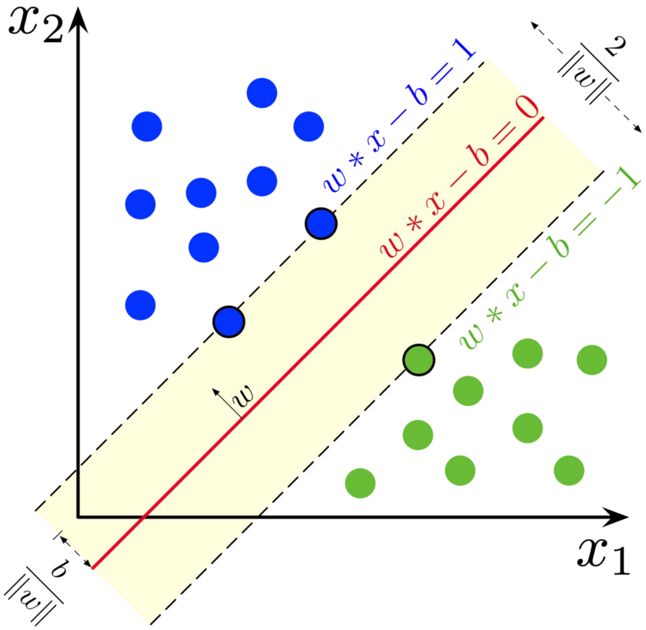
Maximum margin classifier

$$\boxed{\mathbf{w}^{\mathsf{T}}\mathbf{x} - b = 0,}$$

Linear SVM: b, $\mathbf{w}$.

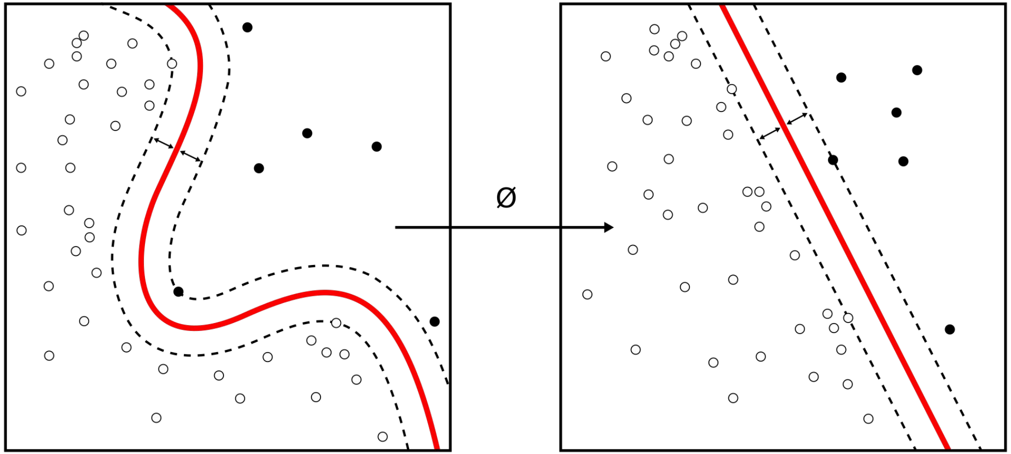
Margin : $\frac{2}{\|\mathbf{w}\|}$,

Maximize



Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors.(Wiki)

Nonlinearity through Kernels



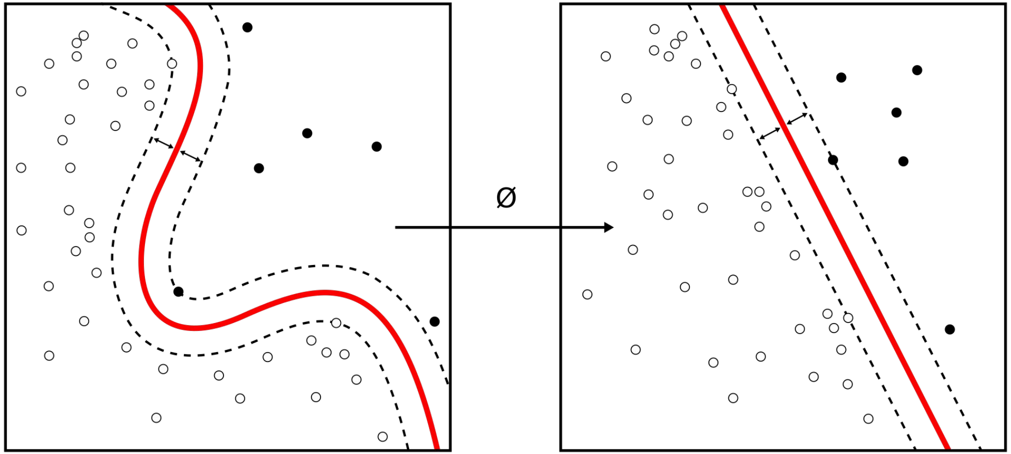
Kernel Machine(Wiki)

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

Nonlinearity through Kernels

$$
\begin{aligned}
k(\mathbf{x}, \mathbf{z}) &= \left(\mathbf{x}^{\mathrm{T}}\mathbf{z}\right)^2 = (x_1 z_1 + x_2 z_2)^2 \\
&= x_1^2 z_1^2 + 2x_1 z_1 x_2 z_2 + x_2^2 z_2^2 \\
&= (x_1^2, \sqrt{2}x_1 x_2, x_2^2)(z_1^2, \sqrt{2}z_1 z_2, z_2^2)^{\mathrm{T}} \\
&= \phi(\mathbf{x})^{\mathrm{T}}\phi(\mathbf{z}).
\end{aligned}
$$

Polynomial kernel



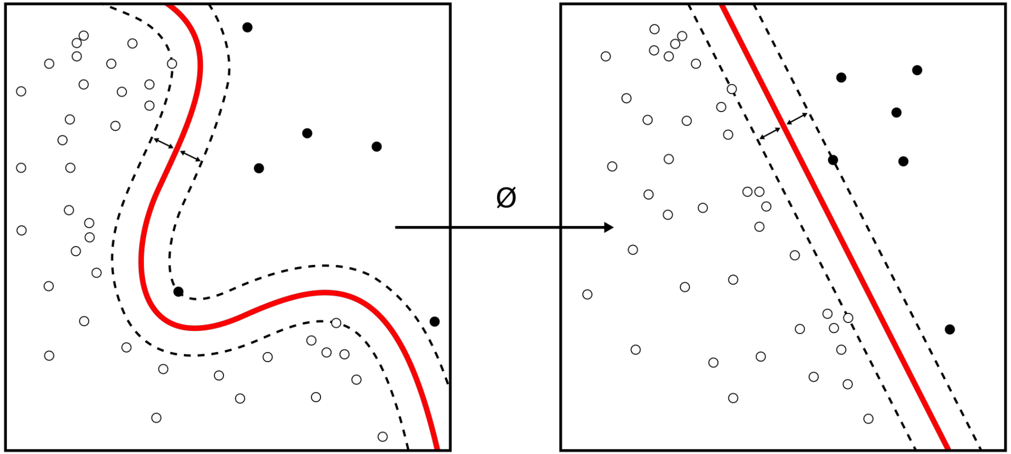
Kernel Machine(Wiki)

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

Nonlinearity through Kernels

$$
\begin{aligned}
k(\mathbf{x}, \mathbf{z}) &= \left(\mathbf{x}^{\mathrm{T}}\mathbf{z}\right)^2 = (x_1 z_1 + x_2 z_2)^2 \\
&= x_1^2 z_1^2 + 2 x_1 z_1 x_2 z_2 + x_2^2 z_2^2 \\
&= (x_1^2, \sqrt{2} x_1 x_2, x_2^2)(z_1^2, \sqrt{2} z_1 z_2, z_2^2)^{\mathrm{T}} \\
&= \phi(\mathbf{x})^{\mathrm{T}} \phi(\mathbf{z}).
\end{aligned}
$$

Polynomial kernel



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

Nonlinearity through Kernels

Some common kernels include:

- Polynomial (homogeneous): $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^d$. Particularly, when $d = 1$, this becomes the linear kernel.
- Polynomial (inhomogeneous): $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + r)^d$.
- Gaussian radial basis function: $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$ for $\gamma > 0$. Sometimes parametrized using $\gamma = 1/(2\sigma^2)$.
- Sigmoid function (Hyperbolic tangent): $k(\mathbf{x_i}, \mathbf{x_j}) = \tanh(\kappa \mathbf{x}_i \cdot \mathbf{x}_j + c)$ for some (not every) $\kappa > 0$ and $c < 0$.

The kernel is related to the transform $\varphi(\mathbf{x}_i)$ by the equation $k(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x_j})$. The value $\mathbf{w}$ is also in the transformed space, with $\mathbf{w} = \sum_i \alpha_i y_i \varphi(\mathbf{x}_i)$. Dot products with $\mathbf{w}$ for classification can again be computed by the kernel trick, i.e. $\mathbf{w} \cdot \varphi(\mathbf{x}) = \sum_i \alpha_i y_i k(\mathbf{x}_i, \mathbf{x})$.

Kernel Machine(Wiki)