

# Chapter ML:I

## I. Introduction

- ❑ Examples of Learning Tasks
- ❑ Specification of Learning Tasks
- ❑ Elements of Machine Learning
- ❑ Notation Overview
- ❑ Classification Approaches Overview

# Notation Overview

## Data, Sets, and Distributions

Symbol	Semantics
$x, x_i, x_1, \dots, x_p$	Feature
$\mathbf{x} = (x_1, \dots, x_p)^T \in \mathbf{R}^p$	Feature vector
$\mathbf{x} = (1, x_1, \dots, x_p)^T \in \mathbf{R}^{p+1}$ , i.e., $x_0 = 1$	Extended feature vector
$\mathbf{X}$	Feature space, Cartesian product of the domains of the $p$ dimensions of a feature vector $\mathbf{x}$ .
$X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$	Multiset of feature vectors
$X$	Random variable (randomness regarding feature $x$ of an object $o$ )
$\mathbf{X}$	Multivariate random variable, random vector (randomness regarding feature vector $\mathbf{x}$ of an object $o$ )

# Notation Overview

## Indexing

Running	Sequence	Semantics of maximum
$\varphi_s$	$\in \{\varphi_1, \dots, \varphi_d\}$	Number of layers in a multilayer perceptron
$\varphi_i$	$\in \{\varphi_1, \dots, \varphi_k\}$	Number of classes Number of folds during cross validation
$\varphi_l$	$\in \{\varphi_1, \dots, \varphi_m\}$	Number of elements in a domain of a feature Number of hyperparameter values during model selection
$\varphi_i$	$\in \{\varphi_1, \dots, \varphi_n\}$	Number of elements in a data set $D$
$\varphi_j$	$\in \{\varphi_1, \dots, \varphi_p\}$	Dimension of a feature space or a feature vector

# Notation Overview

## Functions

Function definition	Function name	Occurrence
$I_{\neq}(a, b) = \begin{cases} 0 & a = b \\ 1 & a \neq b \end{cases}$	Indicator function	Part II: Machine Learning Basics Part III: Linear Models
$f(x) = \dots$	function	Part :

# Notation Overview

## Algorithms

Signature	Algorithm name	Occurrence
$LMS(D, \eta)$	Least Mean Squares	Part I: Introduction Examples of Learning Tasks
$ALG(\dots)$	algorithm	Part : ...

# Classification Approaches Overview

				Search in hypothesis space																	
Taxonomy		Model function	Classification rule	Optimization principle	Optimization objective (loss/cost function [+ regularization])		Optimization approach (algorithm)														
Classification approaches	Discriminative approaches	Linear decision boundary (in inner product space)	Linear decision boundary in input space	Perceptron: $y(\mathbf{x}) = \text{heaviside}(\mathbf{w}^T \mathbf{x})$	$\mathbf{w}^T \mathbf{x} \begin{cases} \geq 0 \\ < 0 \end{cases}$	$\mathbf{w}^T = (w_0, \dots, w_p)$ $x_0 = 1$	Exploit misclassified examples individually: Hebbian learning	$\sim$	No misclassified example	$\sim$	Perceptron training algorithm										
				Linear function: $y(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$							Linear regression	+ Regularization	Squared loss (residual sum of squares, RSS)	+ $L_1$ or $L_2$ norm on $\mathbf{w} _{1, \dots, p}$	Gradient descent: – batch – incremental – stochastic						
				Logistic function: $y(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{x}}}$												Logistic regression	Logistic loss (derived via ML)	Newton-Raphson, BFGS			
				SVM w/o kernel (aka linear kernel)															Empirical risk minimization	Regularized hinge loss	Quadratic prog., sub-grad. descent
		Nonlinear in input / linear in feature space		$y(\mathbf{x}) = \text{sign}(\mathbf{w}^T \phi(\mathbf{x}))$	$\mathbf{w}^T \phi(\mathbf{x}) \begin{cases} \geq 0 \\ < 0 \end{cases}$	$\mathbf{w}^T = (w_0, \dots, w_{ \mathbf{w} })$ $\phi_0(\mathbf{x}) = 1$	Linear regression (nonlinear in predictors)	+ Regularization	Empirical risk minimization	$\sim$	Squared loss	+ $L_1$ or $L_2$ norm on $\mathbf{w} _{1, \dots,  \mathbf{w} }$	Gradient descent: – batch – incremental – stochastic								
				$y(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T \phi(\mathbf{x})}}$										Log. regression (nonlinear in predictors)	Logistic loss	Regularized hinge loss	Backpropagation algorithm				
				SVM with nonlinear kernel														Empirical risk minimization	Regularized hinge loss	Quadratic prog., sub-grad. descent	
Unrestricted decision boundary	Polythetic		Multilayer percep.: $\mathbf{y}(\mathbf{x}) = \sigma(W^0(\sigma^1(W^{\text{h}}\mathbf{x})))$	$\text{argmax}_{c \in C} \{ y_c(\mathbf{x}) \}$	Regression	Squared loss (residual sum of squares, RSS)	$\sim$	No misclassified example	$\sim$	Candidate elimination algorithm											
Monothetic feature analysis			Nominal feat. $\bigwedge_i x_i = v_i$ $i = 1, \dots, p$	Test if $\mathbf{x}$ is a model for $\alpha$ (= fulfills $\alpha$ ). $\alpha$ is a formula in DNF.	Maximize version space	No misclassified example	$\sim$	0/1 Loss (= number of misclassified examples)	+ Tree height, external path length	$\sim$	Algorithms: ID3, C4.5, C5.0, CART										
			$\bigvee_i \bigwedge_j x_{ij} = v_{ij}$ $i = 1, \dots,  \text{leaves} $ $j = 1, \dots, \text{depth}(l_i)$																		
			Arbitrary features: DNF ( $\bigvee_i \bigwedge_j$ ) on domain predicates																		
Generative approaches	Statistical approaches			Bayes rule for combined conditional events	$\text{argmax}_{c \in C} \{ \text{Naive Bayes probabilities} \}$	Decision tree: (greedy) feature-wise splitting of example set	+ Regularization	0/1 Loss (= number of misclassified examples)	+ Tree height, external path length	$\sim$	(exhaustive) search in space of domain splittings										
				$X \sim N(\mu, \sigma^2)$ (or other family)																	
				Maximum a-posteriori hypothesis	$\text{argmax}_{c \in C} \{ P(\mathbf{x}   \mu_c, \sigma_c) \}$	$\sim$ Goodness of fit, e.g. according to chi-squared, Kolmogorov-Smirnov															