

Chapter ML:I (continued)

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 |
|-------------|---|---|
| φ_s | $\in \{ \varphi_1, \dots, \varphi_d \}$ | Number of layers in a multilayer perceptron |
| φ_i | $\in \{ \varphi_1, \dots, \varphi_k \}$ | Number of classes Number of folds during cross validation |
| φ_l | $\in \{ \varphi_1, \dots, \varphi_m \}$ | Number of elements in a domain of a feature Number of hyperparameter values during model selection |
| φ_i | $\in \{ \varphi_1, \dots, \varphi_n \}$ | Number of elements in a data set D |
| φ_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 | ~ | No misclassified example | ~ | Perceptron training algorithm | | | |
| | | | | Linear function: $y(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$ | | | | | | | | | | |
| | | | | Logistic function: $y(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{x}}}$ | | | | | | | | | | |
| | | | | SVM w/o kernel (aka linear kernel) | | | | | | | | | | |
| | | | | $y(\mathbf{x}) = \text{sign}(\mathbf{w}^T \phi(\mathbf{x}))$ | | | | | | | | | | |
| | | Nonlinear in input / linear in feature space | $y(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T \phi(\mathbf{x})}}$ | $\mathbf{w}^T = (w_0, \dots, w_{ \mathbf{w} })$ $\phi_0(\mathbf{x}) = 1$ | Linear regression (nonlinear in predictors) Log. regression (nonlinear in predictors) | + Regularization | Squared loss Logistic loss | L_1 or L_2 norm on $\mathbf{w}_{ 1, \dots, \mathbf{w} }$ | Gradient descent: – batch – incremental – stochastic Newton-Raphson, BFGS | | | | | |
| | | | SVM with nonlinear kernel | Empirical risk minimization | | | | | | Regularized hinge loss | Quadratic prog., sub-grad. descent | | | |
| | | | Multilayer percep.: $\mathbf{y}(\mathbf{x}) = \sigma(W^0(\sigma^1(W^h \mathbf{x})))$ | $\text{argmax}_{c \in C} \{y_c(\mathbf{x})\}$ | | | | | | Regression | Squared loss (residual sum of squares, RSS) | Backpropagation algorithm | | |
| | | | Unrestricted decision boundary | Polythetic | | | | | | Nominal feat. $\bigwedge_i x_i = v_i$ $i = 1, \dots, p$ $\bigvee_i \bigwedge_j x_{ij} = v_{ij}$ $i = 1, \dots, \text{leaves} $ $j = 1, \dots, \text{depth}(l_i)$ | Test if \mathbf{x} is a model for α (= fulfills α). α is a formula in DNF. | Maximize version space | No misclassified example | Candidate elimination algorithm |
| | | | | | | | | | | | | | | |
| Generative approaches | Statistical approaches | Bayes rule for combined conditional events | $\text{argmax}_{c \in C} \{ \text{Naive Bayes probabilities} \}$ | Maximum a-posteriori hypothesis | ~ | Goodness of fit, e.g. according to chi-squared, Kolmogorov-Smirnov | | | | | | | | |
| | | $X \sim N(\mu, \sigma^2)$ (or other family) | $\text{argmax}_{c \in C} \{P(\mathbf{x} \mu_c, \sigma_c)\}$ | | | | | | | | | | | |