

Preface

Use the template *preface.tex* together with the Springer document class SVMono (monograph-type books) or SVMult (edited books) to style your preface in the Springer layout.

A preface is a book's preliminary statement, usually written by the *author or editor* of a work, which states its origin, scope, purpose, plan, and intended audience, and which sometimes includes afterthoughts and acknowledgments of assistance.

When written by a person other than the author, it is called a foreword. The preface or foreword is distinct from the introduction, which deals with the subject of the work.

Customarily *acknowledgments* are included as last part of the preface.

Place(s),
month year

Firstname Surname
Firstname Surname

Contents

Contents	iii	1.3 Some basic concepts	3
Notation	ix	1.3.1 Parametric vs non-parametric models	3
1 Introduction	1	1.3.2 A simple non-parametric classifier: K-nearest neighbours	3
1.1 Types of machine learning	1	1.3.3 Overfitting	3
1.2 Three elements of a machine learning model	1	1.3.4 Cross validation	3
1.2.1 Representation	1	1.3.5 Model selection	4
1.2.2 Evaluation	1	Glossary	5
1.2.3 Optimization	3		

List of Contributors

Firstname Surname

ABC Institute, 123 Prime Street, Daisy Town, NA 01234, USA, e-mail: smith@smith.edu

Firstname Surname

XYZ Institute, Technical University, Albert-Schweitzer-Str. 34, 1000 Berlin, Germany, e-mail: meier@tu.edu

Acronyms

Use the template *acronym.tex* together with the Springer document class *SVMono* (monograph-type books) or *SVMult* (edited books) to style your list(s) of abbreviations or symbols in the Springer layout.

Lists of abbreviations, symbols and the like are easily formatted with the help of the Springer-enhanced `description` environment.

ABC	Spelled-out abbreviation and definition
BABI	Spelled-out abbreviation and definition
CABR	Spelled-out abbreviation and definition

Notation

Introduction

It is very difficult to come up with a single, consistent notation to cover the wide variety of data, models and algorithms that we discuss. Furthermore, conventions differ between machine learning and statistics, and between different books and papers. Nevertheless, we have tried to be as consistent as possible. Below we summarize most of the notation used in this book, although individual sections may introduce new notation. Note also that the same symbol may have different meanings depending on the context, although we try to avoid this where possible.

General math notation

Symbol	Meaning
$\lfloor x \rfloor$	Floor of x , i.e., round down to nearest integer
$\lceil x \rceil$	Ceiling of x , i.e., round up to nearest integer
$\mathbf{x} \otimes \mathbf{y}$	Convolution of \mathbf{x} and \mathbf{y}
$\mathbf{x} \odot \mathbf{y}$	Hadamard (elementwise) product of \mathbf{x} and \mathbf{y}
$a \wedge b$	logical AND
$a \vee b$	logical OR
$\neg a$	logical NOT
$\mathbb{I}(x)$	Indicator function, $\mathbb{I}(x) = 1$ if x is true, else $\mathbb{I}(x) = 0$
∞	Infinity
\rightarrow	Tends towards, e.g., $n \rightarrow \infty$
\propto	Proportional to, so $y = ax$ can be written as $y \propto x$
$ x $	Absolute value
$ \mathcal{S} $	Size (cardinality) of a set
$n!$	Factorial function
∇	Vector of first derivatives
∇^2	Hessian matrix of second derivatives
\triangleq	Defined as
$O(\cdot)$	Big-O: roughly means order of magnitude
\mathbb{R}	The real numbers
$1:n$	Range (Matlab convention): $1:n = 1, 2, \dots, n$
\approx	Approximately equal to
$\arg \max_x f(x)$	Argmax: the value x that maximizes f
$B(a, b)$	Beta function, $B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$
$B(\alpha)$	Multivariate beta function, $\frac{\prod_k \Gamma(\alpha_k)}{\Gamma(\sum_k \alpha_k)}$
$\binom{n}{k}$	n choose k , equal to $n!/(k!(n-k)!)$
$\delta(x)$	Dirac delta function, $\delta(x) = \infty$ if $x = 0$, else $\delta(x) = 0$
$\exp(x)$	Exponential function e^x
$\Gamma(x)$	Gamma function, $\Gamma(x) = \int_0^\infty u^{x-1} e^{-u} du$
$\Psi(x)$	Digamma function, $\Psi(x) = \frac{d}{dx} \log \Gamma(x)$

\mathcal{X}	A set from which values are drawn (e.g., $\mathcal{X} = \mathbb{R}^D$)
---------------	---

Linear algebra notation

We use boldface lower-case to denote vectors, such as \mathbf{x} , and boldface upper-case to denote matrices, such as \mathbf{X} . We denote entries in a matrix by non-bold upper case letters, such as X_{ij} .

Vectors are assumed to be column vectors, unless noted otherwise. We use (x_1, \dots, x_D) to denote a column vector created by stacking D scalars. If we write $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$, where the left hand side is a matrix, we mean to stack the \mathbf{x}_i along the columns, creating a matrix.

Symbol	Meaning
$\mathbf{X} \succ 0$	\mathbf{X} is a positive definite matrix
$tr(\mathbf{X})$	Trace of a matrix
$det(\mathbf{X})$	Determinant of matrix \mathbf{X}
$ \mathbf{X} $	Determinant of matrix \mathbf{X}
\mathbf{X}^{-1}	Inverse of a matrix
\mathbf{X}^\dagger	Pseudo-inverse of a matrix
\mathbf{X}^T	Transpose of a matrix
\mathbf{x}^T	Transpose of a vector
$diag(\mathbf{x})$	Diagonal matrix made from vector \mathbf{x}
$diag(\mathbf{X})$	Diagonal vector extracted from matrix \mathbf{X}
\mathbf{I} or \mathbf{I}_d	Identity matrix of size $d \times d$ (ones on diagonal, zeros of)
$\mathbf{1}$ or $\mathbf{1}_d$	Vector of ones (of length d)
$\mathbf{0}$ or $\mathbf{0}_d$	Vector of zeros (of length d)
$\ \mathbf{x}\ = \ \mathbf{x}\ _2$	Euclidean or ℓ_2 norm $\sqrt{\sum_{j=1}^d x_j^2}$
$\ \mathbf{x}\ _1$	ℓ_1 norm $\sum_{j=1}^d x_j $
$\mathbf{X}_{:,j}$	j 'th column of matrix
$\mathbf{X}_{i,:}$	transpose of i 'th row of matrix (a column vector)
$\mathbf{X}_{i,j}$	Element (i, j) of matrix \mathbf{X}
$\mathbf{x} \otimes \mathbf{y}$	Tensor product of \mathbf{x} and \mathbf{y}

Probability notation

We denote random and fixed scalars by lower case, random and fixed vectors by bold lower case, and random and fixed matrices by bold upper case. Occasionally we use non-bold upper case to denote scalar random variables. Also, we use $p()$ for both discrete and continuous random variables

Symbol	Meaning
X, Y	Random variable
$P()$	Probability of a random event
$F()$	Cumulative distribution function(CDF), also called distribution function
$p(x)$	Probability mass function(PMF)
$f(x)$	probability density function(PDF)
$F(x, y)$	Joint CDF
$p(x, y)$	Joint PMF
$f(x, y)$	Joint PDF

$p(X Y)$	Conditional PMF, also called conditional probability
$f_{X Y}(x y)$	Conditional PDF
$X \perp Y$	X is independent of Y
$X \not\perp Y$	X is not independent of Y
$X \perp Y Z$	X is conditionally independent of Y given Z
$X \not\perp Y Z$	X is not conditionally independent of Y given Z
$X \sim p$	X is distributed according to distribution p
α	Parameters of a Beta or Dirichlet distribution
$\text{cov}[X]$	Covariance of X
$\mathbb{E}[X]$	Expected value of X
$\mathbb{E}_q[X]$	Expected value of X wrt distribution q
$\mathbb{H}(X)$ or $\mathbb{H}(p)$	Entropy of distribution $p(X)$
$\mathbb{I}(X;Y)$	Mutual information between X and Y
$\mathbb{KL}(p q)$	KL divergence from distribution p to q
$\ell(\theta)$	Log-likelihood function
$L(\theta, a)$	Loss function for taking action a when true state of nature is θ
λ	Precision (inverse variance) $\lambda = 1/\sigma^2$
Λ	Precision matrix $\Lambda = \Sigma^{-1}$
$\text{mode}[\mathbf{X}]$	Most probable value of \mathbf{X}
μ	Mean of a scalar distribution
$\boldsymbol{\mu}$	Mean of a multivariate distribution
Φ	cdf of standard normal
ϕ	pdf of standard normal
π	multinomial parameter vector, Stationary distribution of Markov chain
ρ	Correlation coefficient
$\text{sigm}(x)$	Sigmoid (logistic) function, $\frac{1}{1 + e^{-x}}$
σ^2	Variance
Σ	Covariance matrix
$\text{var}[x]$	Variance of x
ν	Degrees of freedom parameter
Z	Normalization constant of a probability distribution

Machine learning/statistics notation

In general, we use upper case letters to denote constants, such as C, K, M, N, T , etc. We use lower case letters as dummy indexes of the appropriate range, such as $c = 1 : C$ to index classes, $i = 1 : M$ to index data cases, $j = 1 : N$ to index input features, $k = 1 : K$ to index states or clusters, $t = 1 : T$ to index time, etc.

We use x to represent an observed data vector. In a supervised problem, we use y or \mathbf{y} to represent the desired output label. We use z to represent a hidden variable. Sometimes we also use q to represent a hidden discrete variable.

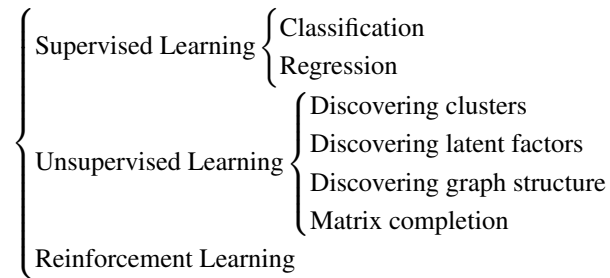
Symbol	Meaning
C	Number of classes
D	Dimensionality of data vector (number of features)
N	Number of data cases
N_c	Number of examples of class c , $N_c = \sum_{i=1}^N \mathbb{I}(y_i = c)$
R	Number of outputs (response variables)
\mathcal{D}	Training data $\mathcal{D} = \{(\mathbf{x}_i, y_i) i = 1 : N\}$
\mathcal{D}_{test}	Test data
\mathcal{X}	Input space
\mathcal{Y}	Output space

K	Number of states or dimensions of a variable (often latent)
$k(x, y)$	Kernel function
\mathbf{K}	Kernel matrix
\mathcal{H}	Hypothesis space
L	Loss function
$J(\boldsymbol{\theta})$	Cost function
$f(\mathbf{x})$	Decision function
$P(y \mathbf{x})$	TODO
λ	Strength of ℓ_2 or ℓ_1 <i>regularizer</i>
$\phi(x)$	Basis function expansion of feature vector \mathbf{x}
Φ	Basis function expansion of design matrix \mathbf{X}
$q()$	Approximate or proposal distribution
$Q(\boldsymbol{\theta}, \boldsymbol{\theta}_{old})$	Auxiliary function in EM
T	Length of a sequence
$T(\mathcal{D})$	Test statistic for data
\mathbf{T}	Transition matrix of Markov chain
$\boldsymbol{\theta}$	Parameter vector
$\boldsymbol{\theta}^{(s)}$	s 'th sample of parameter vector
$\hat{\boldsymbol{\theta}}$	Estimate (usually MLE or MAP) of $\boldsymbol{\theta}$
$\hat{\boldsymbol{\theta}}_{MLE}$	Maximum likelihood estimate of $\boldsymbol{\theta}$
$\hat{\boldsymbol{\theta}}_{MAP}$	MAP estimate of $\boldsymbol{\theta}$
$\bar{\boldsymbol{\theta}}$	Estimate (usually posterior mean) of $\boldsymbol{\theta}$
\mathbf{w}	Vector of regression weights (called $\boldsymbol{\beta}$ in statistics)
b	intercept (called ϵ in statistics)
\mathbf{W}	Matrix of regression weights
x_{ij}	Component (i.e., feature) j of data case i , for $i = 1 : N, j = 1 : D$
\mathbf{x}_i	Training case, $i = 1 : N$
\mathbf{X}	Design matrix of size $N \times D$
$\bar{\mathbf{x}}$	Empirical mean $\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$
$\tilde{\mathbf{x}}$	Future test case
\mathbf{x}_*	Feature test case
\mathbf{y}	Vector of all training labels $\mathbf{y} = (y_1, \dots, y_N)$
z_{ij}	Latent component j for case i

Chapter 1

Introduction

1.1 Types of machine learning



1.2 Three elements of a machine learning model

Model = Representation + Evaluation + Optimization¹

1.2.1 Representation

In supervised learning, a model must be represented as a conditional probability distribution $P(y|x)$ (usually we call it classifier) or a decision function $f(x)$. The set of classifiers (or decision functions) is called the hypothesis space of the model. Choosing a representation for a model is tantamount to choosing the hypothesis space that it can possibly learn.

1.2.2 Evaluation

In the hypothesis space, an evaluation function (also called objective function or risk function) is needed to distinguish good classifiers (or decision functions) from bad ones.

1.2.2.1 Loss function and risk function

Definition 1.1. In order to measure how well a function fits the training data, a **loss function** $L : Y \times Y \rightarrow R \geq 0$ is defined. For training example (x_i, y_i) , the loss of predicting the value \hat{y} is $L(y_i, \hat{y})$.

The following is some common loss functions:

1. 0-1 loss function

$$L(Y, f(X)) = \mathbb{I}(Y, f(X)) = \begin{cases} 1, & Y \neq f(X) \\ 0, & Y = f(X) \end{cases}$$

2. Quadratic(squared) loss function $L(Y, f(X)) = \frac{1}{2} (Y - f(X))^2$

3. Absolute loss function $L(Y, f(X)) = |Y - f(X)|$

¹ Domingos, P. A few useful things to know about machine learning. Commun. ACM. 55(10):7887 (2012).

4. Exponential loss function $L(Y, f(X)) = \exp(-\hat{y}_i f(\mathbf{x}_i))$

5. Logarithmic loss function

$$L(Y, P(Y|X)) = -\log P(Y|X)$$

Name	Loss	Derivative	f^*	Algorithm
Squared error	$\frac{1}{2}(y_i - f(\mathbf{x}_i))^2$	$y_i - f(\mathbf{x}_i)$	$\mathbb{E}[y \mathbf{x}_i]$	L2Boosting
Absolute error	$ y_i - f(\mathbf{x}_i) $	$\text{sgn}(y_i - f(\mathbf{x}_i))$	$\text{median}(y \mathbf{x}_i)$	Gradient boosting
Exponential loss	$\exp(-\hat{y}_i f(\mathbf{x}_i))$	$-\hat{y}_i \exp(-\hat{y}_i f(\mathbf{x}_i))$	$\frac{1}{2} \log \frac{\pi_i}{1-\pi_i}$	AdaBoost
Logloss	$\log(1 + e^{-\hat{y}_i f_i})$	$y_i - \pi_i$	$\frac{1}{2} \log \frac{\pi_i}{1-\pi_i}$	LogitBoost

Definition 1.2. The risk of function f is defined as the expected loss of f :

$$R_{\text{exp}}(f) = E[L(Y, f(X))] = \int L(y, f(x)) P(x, y) dx dy \quad (1.1)$$

which is also called expected loss or **risk function**.

Definition 1.3. The risk function $R_{\text{exp}}(f)$ can be estimated from the training data as

$$R_{\text{emp}}(f) = \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) \quad (1.2)$$

which is also called empirical loss or **empirical risk**.

You can define your own loss function, but if you're a novice, you're probably better off using one from the literature. There are conditions that loss functions should meet²:

1. They should approximate the actual loss you're trying to minimize. As was said in the other answer, the standard loss functions for classification is zero-one-loss (misclassification rate) and the ones used for training classifiers are approximations of that loss.
2. The loss function should work with your intended optimization algorithm. That's why zero-one-loss is not used directly: it doesn't work with gradient-based optimization methods since it doesn't have a well-defined gradient (or even a subgradient, like the hinge loss for SVMs has).

The main algorithm that optimizes the zero-one-loss directly is the old perceptron algorithm(chapter §??).

1.2.2.2 ERM and SRM

Definition 1.4. ERM(Empirical risk minimization)

$$\min_{f \in \mathcal{F}} R_{\text{emp}}(f) = \min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) \quad (1.3)$$

Definition 1.5. Structural risk

$$R_{\text{smp}}(f) = \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) + \lambda J(f) \quad (1.4)$$

Definition 1.6. SRM(Structural risk minimization)

$$\min_{f \in \mathcal{F}} R_{\text{srm}}(f) = \min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) + \lambda J(f) \quad (1.5)$$

² <http://t.cn/zTrDxLO>

1.2.3 Optimization

Finally, we need a **training algorithm**(also called **learning algorithm**) to search among the classifiers in the hypothesis space for the highest-scoring one. The choice of optimization technique is key to the **efficiency** of the model.

1.3 Some basic concepts

1.3.1 Parametric vs non-parametric models

1.3.2 A simple non-parametric classifier: K-nearest neighbours

1.3.2.1 Representation

$$y = f(\mathbf{x}) = \arg \min_c \sum_{\mathbf{x}_i \in N_k(\mathbf{x})} \mathbb{I}(y_i = c) \quad (1.6)$$

where $N_k(\mathbf{x})$ is the set of k points that are closest to point \mathbf{x} .

Usually use **k-d tree** to accelerate the process of finding k nearest points.

1.3.2.2 Evaluation

No training is needed.

1.3.2.3 Optimization

No training is needed.

1.3.3 Overfitting

1.3.4 Cross validation

Definition 1.7. Cross validation, sometimes called *rotation estimation*, is a *model validation* technique for assessing how the results of a statistical analysis will generalize to an independent data set³.

Common types of cross-validation:

1. K-fold cross-validation. In k-fold cross-validation, the original sample is randomly partitioned into k equal size subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k - 1 subsamples are used as training data.
2. 2-fold cross-validation. Also, called simple cross-validation or holdout method. This is the simplest variation of k-fold cross-validation, k=2.
3. Leave-one-out cross-validation(*LOOCV*). k=M, the number of original samples.

³ [http://en.wikipedia.org/wiki/Cross-validation_\(statistics\)](http://en.wikipedia.org/wiki/Cross-validation_(statistics))

1.3.5 Model selection

When we have a variety of models of different complexity (e.g., linear or logistic regression models with different degree polynomials, or KNN classifiers with different values of K), how should we pick the right one? A natural approach is to compute the **misclassification rate** on the training set for each method.

Glossary

Use the template *glossary.tex* together with the Springer document class SVMono (monograph-type books) or SVMult (edited books) to style your glossary in the Springer layout.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.