Chapman & Hall/CRC
Machine Learning & Pattern Recognition Series

Ensemble Methods

Foundations and Algorithms




Zhi-Hua Zhou

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

Ensemble Methods

Foundations and Algorithms

Chapman & Hall/CRC
Machine Learning & Pattern Recognition Series

SERIES EDITORS

Ralf Herbrich and Thore Graepel
Microsoft Research Ltd.
Cambridge, UK

AIMS AND SCOPE

This series reflects the latest advances and applications in machine learning and pattern recognition through the publication of a broad range of reference works, textbooks, and handbooks. The inclusion of concrete examples, applications, and methods is highly encouraged. The scope of the series includes, but is not limited to, titles in the areas of machine learning, pattern recognition, computational intelligence, robotics, computational/statistical learning theory, natural language processing, computer vision, game AI, game theory, neural networks, computational neuroscience, and other relevant topics, such as machine learning applied to bioinformatics or cognitive science, which might be proposed by potential contributors.

PUBLISHED TITLES

MACHINE LEARNING: An Algorithmic Perspective
Stephen Marsland

HANDBOOK OF NATURAL LANGUAGE PROCESSING,
Second Edition
Nitin Indurkhya and Fred J. Damerau

UTILITY-BASED LEARNING FROM DATA
Craig Friedman and Sven Sandow

A FIRST COURSE IN MACHINE LEARNING
Simon Rogers and Mark Girolami

COST-SENSITIVE MACHINE LEARNING
Balaji Krishnapuram, Shipeng Yu, and Bharat Rao

ENSEMBLE METHODS: FOUNDATIONS AND ALGORITHMS
Zhi-Hua Zhou

Chapman & Hall/CRC
Machine Learning & Pattern Recognition Series

Ensemble Methods

Foundations and Algorithms

Zhi-Hua Zhou

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business
A CHAPMAN & HALL BOOK

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2012 by Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works
Version Date: 20120501

International Standard Book Number-13: 978-1-4398-3005-5 (eBook - PDF)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (http://www.copyright.com/) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

To my parents, wife and son.

Z.-H. Zhou

This page intentionally left blank

Preface

Ensemble methods that train multiple learners and then combine them for use, with Boosting and Bagging as representatives, are a kind of state-of-the-art learning approach. It is well known that an ensemble is usually significantly more accurate than a single learner, and ensemble methods have already achieved great success in many real-world tasks.

It is difficult to trace the starting point of the history of ensemble methods since the basic idea of deploying multiple models has been in use in human society for a long time; however, it is clear that ensemble methods have become a hot topic since the 1990s, and researchers from various fields such as machine learning, pattern recognition, data mining, neural networks and statistics have explored ensemble methods from different aspects.

This book provides researchers, students and practitioners with an introduction to ensemble methods. The book consists of eight chapters which naturally constitute three parts.

Part I is composed of Chapter 1. Though this book is mainly written for readers with a basic knowledge of machine learning and pattern recognition, to enable readers who are unfamiliar with these fields to access the main contents, Chapter 1 presents some "background knowledge" of ensemble methods. It is impossible to provide a detailed introduction to all backgrounds in one chapter, and therefore this chapter serves mainly as a guide to further study. This chapter also serves to explain the terminology used in this book, to avoid confusion caused by other terminologies used in different but relevant fields.

Part II is composed of Chapters 2 to 5 and presents "core knowledge" of ensemble methods. Chapters 2 and 3 introduce Boosting and Bagging, respectively. In addition to algorithms and theories, Chapter 2 introduces multi-class extension and noise tolerance, since classic Boosting algorithms are designed for binary classification, and are usually hurt seriously by noise. Bagging is naturally a multi-class method and less sensitive to noise, and therefore, Chapter 3 does not discuss these issues; instead, Chapter 3 devotes a section to Random Forest and some other random tree ensembles that can be viewed as variants of Bagging. Chapter 4 introduces combination methods. In addition to various averaging and voting schemes, the Stacking method and some other combination methods as well as relevant methods such as mixture of experts are introduced. Chapter 5 focuses on ensemble diversity. After introducing the error-ambiguity and bias-variance

decompositions, many diversity measures are presented, followed by recent advances in information theoretic diversity and diversity generation methods.

Part III is composed of Chapters 6 to 8, and presents "advanced knowledge" of ensemble methods. Chapter 6 introduces ensemble pruning, which tries to prune a trained ensemble to get a better performance. Chapter 7 introduces clustering ensembles, which try to generate better clustering results by combining multiple clusterings. Chapter 8 presents some developments of ensemble methods in semi-supervised learning, active learning, cost-sensitive learning and class-imbalance learning, as well as comprehensibility enhancement.

It is not the goal of the book to cover all relevant knowledge of ensemble methods. Ambitious readers may be interested in *Further Reading* sections for further information.

Two other books [Kuncheva, 2004, Rokach, 2010] on ensemble methods have been published before this one. To reflect the fast development of this field, I have attempted to present an updated and in-depth overview. However, when writing this book, I found this task more challenging than expected. Despite abundant research on ensemble methods, a thorough understanding of many essentials is still needed, and there is a lack of thorough empirical comparisons of many technical developments. As a consequence, several chapters of the book simply introduce a number of algorithms, while even for chapters with discussions on theoretical issues, there are still important yet unclear problems. On one hand, this reflects the still developing situation of the ensemble methods field; on the other hand, such a situation provides a good opportunity for further research.

The book could not have been written, at least not in its current form, without the help of many people. I am grateful to Tom Dietterich who has carefully read the whole book and given very detailed and insightful comments and suggestions. I want to thank Songcan Chen, Nan Li, Xu-Ying Liu, Fabio Roli, Jianxin Wu, Yang Yu and Min-Ling Zhang for helpful comments. I also want to thank Randi Cohen and her colleagues at Chapman & Hall/CRC Press for cooperation.

Last, but definitely not least, I am indebted to my family, friends and students for their patience, support and encouragement.

Zhi-Hua Zhou
Nanjing, China

Notations

x	variable
$\boldsymbol{x}$	vector
$\mathbf{A}$	matrix
$\mathbf{I}$	identity matrix
$\mathcal{X}, \mathcal{Y}$	input and output spaces
$\mathcal{D}$	probability distribution
D	data sample (data set)
$\mathcal{N}$	normal distribution
$\mathcal{U}$	uniform distribution
$\mathcal{H}$	hypothesis space
H	set of hypotheses
$h(\cdot)$	hypothesis (learner)
$\mathfrak{L}$	learning algorithm
$p(\cdot)$	probability density function
$p(\cdot \mid \cdot)$	conditional probability density function
$P(\cdot)$	probability mass function
$P(\cdot \mid \cdot)$	conditional probability mass function
$\mathbb{E}_{\cdot \sim \mathcal{D}}[f(\cdot)]$	mathematical expectation of function $f(\cdot)$ to $\cdot$ under distribution $\mathcal{D}$. $\mathcal{D}$ and/or $\cdot$ is ignored when the meaning is clear
$var_{\cdot \sim \mathcal{D}}[f(\cdot)]$	variance of function $f(\cdot)$ to $\cdot$ under distribution $\mathcal{D}$
$\mathbb{I}(\cdot)$	indicator function which takes 1 if $\cdot$ is `true`, and 0 otherwise
$\texttt{sign}(\cdot)$	sign function which takes -1,1 and 0 when $\cdot < 0$, $\cdot > 0$ and $\cdot = 0$, respectively
$err(\cdot)$	error function
$\{\ldots\}$	set
$(\ldots)$	row vector

$(\ldots)^\top$	column vector
$\lvert\cdot\rvert$	size of data set
$\lVert\cdot\rVert$	L_2-norm

Contents

1

Introduction

1.1 Basic Concepts

One major task of machine learning, pattern recognition and data mining is to construct *good* **models** from **data sets**.

A "data set" generally consists of **feature vectors**, where each feature vector is a description of an object by using a set of **features**. For example, take a look at the synthetic *three-Gaussians* data set as shown in Figure 1.1. Here, each object is a data point described by the features x-coordinate, y-coordinate and shape, and a feature vector looks like (.5, .8, cross) or (.4, .5, circle). The number of features of a data set is called **dimension** or **dimensionality**; for example, the dimensionality of the above data set is three. Features are also called **attributes**, a feature vector is also called an **instance**, and sometimes a data set is called a **sample**.

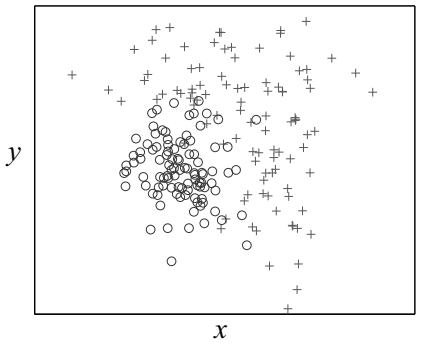


FIGURE 1.1: The synthetic *three-Gaussians* data set.

A "model" is usually a predictive model or a model of the structure of the data that we want to construct or discover from the data set, such as a decision tree, a neural network, a support vector machine, etc. The pro-

cess of generating models from data is called **learning** or **training**, which is accomplished by a **learning algorithm**. The learned model can be called a **hypothesis**, and in this book it is also called a **learner**. There are different learning settings, among which the most common ones are **supervised learning** and **unsupervised learning**. In supervised learning, the goal is to predict the value of a target feature on unseen instances, and the learned model is also called a **predictor**. For example, if we want to predict the shape of the *three-Gaussians* data points, we call "cross" and "circle" **labels**, and the predictor should be able to predict the label of an instance for which the label information is unknown, e.g., (.2, .3). If the label is *categorical*, such as shape, the task is also called **classification** and the learner is also called **classifier**; if the label is *numerical*, such as x-coordinate, the task is also called **regression** and the learner is also called **fitted regression model**. For both cases, the training process is conducted on data sets containing label information, and an instance with known label is also called an **example**. In **binary classification**, generally we use "positive" and "negative" to denote the two class labels. Unsupervised learning does not rely on label information, the goal of which is to discover some inherent distribution information in the data. A typical task is **clustering**, aiming to discover the cluster structure of data points. In most of this book we will focus on supervised learning, especially classification. We will introduce some popular learning algorithms briefly in Section 1.2.

Basically, whether a model is "good" depends on whether it can meet the requirements of the user or not. Different users might have different expectations of the learning results, and it is difficult to know the "right expectation" before the concerned task has been tackled. A popular strategy is to evaluate and estimate the performance of the models, and then let the user to decide whether a model is acceptable, or choose the best available model from a set of candidates. Since the fundamental goal of learning is **generalization**, i.e., being capable of generalizing the "knowledge" learned from training data to unseen instances, a good learner should generalize well, i.e., have a small **generalization error**, also called the **prediction error**. It is infeasible, however, to estimate the generalization error directly, since that requires knowing the **ground-truth** label information which is unknown for unseen instances. A typical empirical process is to let the predictor make predictions on **test data** of which the ground-truth labels are known, and take the **test error** as an estimate of the generalization error. The process of applying a learned model to unseen data is called **testing**. Before testing, a learned model often needs to be configured, e.g., tuning the parameters, and this process also involves the use of data with known ground-truth labels to evaluate the learning performance; this is called **validation** and the data is **validation data**. Generally, the test data should not overlap with the training and validation data; otherwise the estimated performance can be over-optimistic. More introduction on performance evaluation will be given in Section 1.3.

A formal formulation of the learning process is as follows: Denote $\mathcal{X}$ as the instance space, $\mathcal{D}$ as a distribution over $\mathcal{X}$, and f the *ground-truth* target function. Given a training data set $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$, where the instances $\boldsymbol{x}_i$ are drawn ***i.i.d.*** (independently and identically distributed) from $\mathcal{D}$ and $y_i = f(\boldsymbol{x}_i)$, taking classification as an example, the goal is to construct a learner h which minimizes the generalization error

$$err(h) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\mathbb{I}(h(\boldsymbol{x}) \neq f(\boldsymbol{x}))]. \tag{1.1}$$

1.2 Popular Learning Algorithms

1.2.1 Linear Discriminant Analysis

A **linear classifier** consists of a **weight vector** $\boldsymbol{w}$ and a **bias** b. Given an instance $\boldsymbol{x}$, the predicted class label y is obtained according to

$$y = \texttt{sign}(\boldsymbol{w}^\top \boldsymbol{x} + b). \tag{1.2}$$

The classification process is accomplished by two steps. First, the instance space is mapped onto a one-dimensional space (i.e., a line) through the weight vector $\boldsymbol{w}$; then, a point on the line is identified to separate the positive instances from negative ones.

To find the best $\boldsymbol{w}$ and b for separating different classes, a classical linear learning algorithm is *Fisher's linear discriminant analysis* (LDA). Briefly, the idea of LDA is to enable instances of different classes to be far away while instances within the same class to be close; this can be accomplished by making the distance between centers of different classes large while keeping the variance within each class small.

Given a two-class training set, we consider all the positive instances, and obtain the mean $\boldsymbol{\mu}_+$ and the covariance matrix Σ_+; similarly, we consider all the negative instances, and obtain the mean $\boldsymbol{\mu}_-$ and the covariance matrix Σ_-. The distance between the projected class centers is measured as

$$S_B(\boldsymbol{w}) = (\boldsymbol{w}^\top \boldsymbol{\mu}_+ - \boldsymbol{w}^\top \boldsymbol{\mu}_-)^2, \tag{1.3}$$

and the variance within classes is measured as

$$S_W(\boldsymbol{w}) = \boldsymbol{w}^\top \Sigma_+ \boldsymbol{w} + \boldsymbol{w}^\top \Sigma_- \boldsymbol{w}. \tag{1.4}$$

LDA combines these two measures by maximizing

$$J(\boldsymbol{w}) = S_B(\boldsymbol{w})/S_W(\boldsymbol{w}), \tag{1.5}$$

of which the optimal solution has a closed-form

$$\boldsymbol{w}^* = (\Sigma_+ + \Sigma_-)^{-1}(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-). \tag{1.6}$$

After obtaining $\boldsymbol{w}$, it is easy to calculate the bias b. The simplest way is to let b be the middle point between the projected centers, i.e.,

$$b^* = \boldsymbol{w}^\top(\boldsymbol{\mu}_+ + \boldsymbol{\mu}_-)/2, \tag{1.7}$$

which is optimal when the two classes are from normal distributions sharing the same variance.

Figure 1.2 illustrates the decision boundary of an LDA classifier.

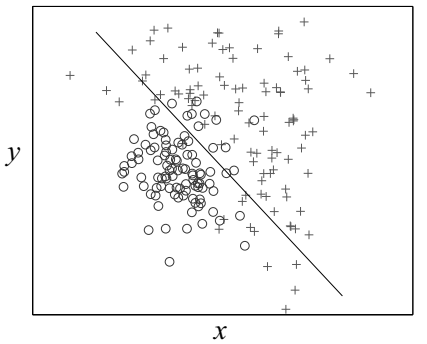


FIGURE 1.2: Decision boundary of LDA on the *three-Gaussians* data set.

1.2.2 Decision Trees

A decision tree consists of a set of tree-structured decision tests working in a *divide-and-conquer* way. Each non-leaf node is associated with a **feature test** also called a **split**; data falling into the node will be split into different subsets according to their different values on the feature test. Each leaf node is associated with a label, which will be assigned to instances falling into this node. In prediction, a series of feature tests is conducted starting from the root node, and the result is obtained when a leaf node is reached. Take Figure 1.3 as an example. The classification process starts by testing whether the value of the feature y-coordinate is larger than 0.73; if so, the instance is classified as "cross", and otherwise the tree tests whether the feature value of x-coordinate is larger than 0.64; if so, the instance is classified as "cross" and otherwise is classified as "circle".

Decision tree learning algorithms are generally recursive processes. In each step, a data set is given and a split is selected, then this split is used to divide the data set into subsets, and each subset is considered as the given data set for the next step. The key of a decision tree algorithm is how to select the splits.

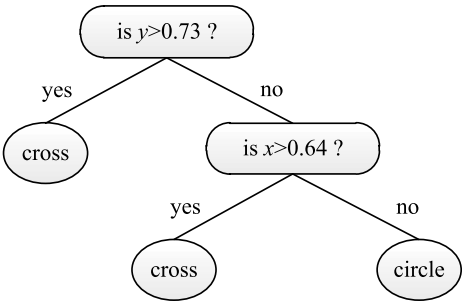


FIGURE 1.3: An example of a decision tree.

In the ID3 algorithm [Quinlan, 1998], the **information gain** criterion is employed for split selection. Given a training set D, the **entropy** of D is defined as

$$Ent(D) = -\sum_{y \in \mathcal{Y}} P(y|D) \log P(y|D). \tag{1.8}$$

If the training set D is divided into subsets $D_1, \ldots, D_k$, the entropy may be reduced, and the amount of the reduction is the information gain, i.e.,

$$G(D; D_1, \ldots, D_k) = Ent(D) - \sum_{i=1}^{k} \frac{|D_k|}{|D|} Ent(D_k). \tag{1.9}$$

Thus, the feature-value pair which will cause the largest information gain is selected for the split.

One problem with the information gain criterion is that features with a lot of possible values will be favored, disregarding their relevance to classification. For example, suppose we are dealing with binary classification and each instance has a unique "*id*", and if the "*id*" is considered as a feature, the information gain of taking this feature as split would be quite large since this split will classify every training instance correctly; however, it cannot generalize and thus will be useless for making prediction on unseen instances.

This deficiency of the information gain criterion is addressed in C4.5 [Quinlan, 1993], the most famous decision tree algorithm. C4.5 employs the **gain ratio**

$$P(D; D_1, \ldots, D_k) = G(D; D_1, \ldots, D_k) \cdot \left(-\sum_{i=1}^{k} \frac{|D_k|}{|D|} \log \frac{|D_k|}{|D|} \right)^{-1}, \tag{1.10}$$

which is a variant of the information gain criterion, taking **normalization** on the number of feature values. In practice, the feature with the highest gain ratio, among features with better-than-average information gains, is selected as the split.

CART [Breiman et al., 1984] is another famous decision tree algorithm, which uses **Gini index** for selecting the split maximizing the Gini

$$G_{gini}(D; D_1, \ldots, D_k) = I(D) - \sum_{i=1}^{k} \frac{|D_k|}{|D|} I(D_k), \tag{1.11}$$

where

$$I(D) = 1 - \sum_{y \in \mathcal{Y}} P(y \mid D)^2. \tag{1.12}$$

It is often observed that a decision tree, which is perfect on the training set, will have a worse generalization ability than a tree which is not-so-good on the training set; this is called **overfitting** which may be caused by the fact that some peculiarities of the training data, such as those caused by noise in collecting training examples, are misleadingly recognized by the learner as the underlying truth. To reduce the risk of overfitting, a general strategy is to employ **pruning** to cut off some tree branches caused by noise or peculiarities of the training set. **Pre-pruning** tries to prune branches when the tree is being grown, while **post-pruning** re-examines fully grown trees to decide which branches should be removed. When a validation set is available, the tree can be pruned according to the validation error: for pre-pruning, a branch will not be grown if the validation error will increase by growing the branch; for post-pruning, a branch will be removed if the removal will decrease the validation error.

Early decision tree algorithms, such as ID3, could only deal with categorical features. Later ones, such as C4.5 and CART, are enabled to deal with numerical features. The simplest way is to evaluate every possible split point on the numerical feature that divides the training set into two subsets, where one subset contains instances with the feature value smaller than the split point while the other subset contains the remaining instances.

When the height of a decision tree is limited to 1, i.e., it takes only one test to make every prediction, the tree is called a **decision stump**. While decision trees are nonlinear classifiers in general, decision stumps are a kind of linear classifiers.

Figure 1.4 illustrates the decision boundary of a typical decision tree.

1.2.3 Neural Networks

Neural networks, also called **artificial neural networks**, originated from simulating biological neural networks. The function of a neural network is determined by the model of **neuron**, the network structure, and the learning algorithm.

Neuron is also called **unit**, which is the basic computational component in neural networks. The most popular neuron model, i.e., the *McCulloch-Pitts* model (**M-P model**), is illustrated in Figure 1.5(a). In this model, input

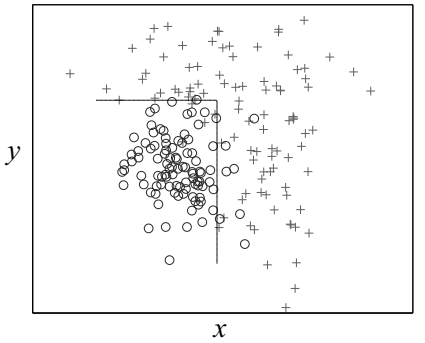


FIGURE 1.4: Decision boundary of a typical decision tree on the *three-Gaussians* data set.

signals are multiplied with corresponding **connection weights** at first, and then signals are aggregated and compared with a threshold, also called **bias** of the neuron. If the aggregated signal is larger than the bias, the neuron will be activated and the output signal is generated by an **activation function**, also called *transfer function* or *squashing function*.

Neurons are linked by weighted connections to form a network. There are many possible network structures, among which the most popular one is the **multi-layer feed-forward network**, as illustrated in Figure 1.5(b). Here the neurons are connected layer-by-layer, and there are neither in-layer connections nor cross-layer connections. There is an **input layer** which receives input feature vectors, where each neuron usually corresponds to one element of the feature vector. The activation function for input neurons is usually set as $f(x) = x$. There is an **output layer** which outputs labels, where each neuron usually corresponds to a possible label, or an element of a *label vector*. The layers between the input and output layers are called **hidden layers**. The hidden neurons and output neurons are functional units, and a popular activation function for them is the **sigmoid function**

$$f(x) = \frac{1}{1 + e^{-x}}. \tag{1.13}$$

Although one may use a network with many hidden layers, the most popular setting is to use one or two hidden layers, since it is known that a feed-forward neural network with one hidden layer is already able to approximate any continuous function, and more complicated algorithms are needed to prevent networks with many hidden layers from suffering from problems such as divergence (i.e., the networks do not converge to a stable state).

The goal of training a neural network is to determine the values of the connection weights and the biases of the neurons. Once these values are

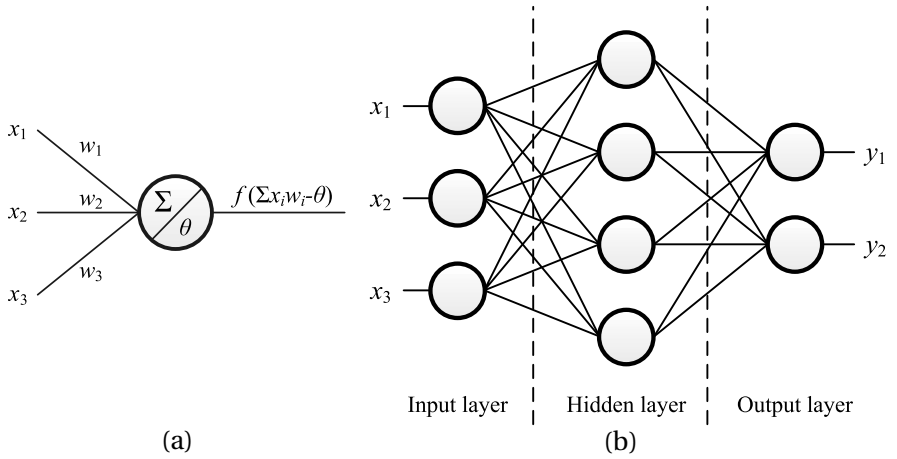


FIGURE 1.5: Illustration of (a) a neuron, and (b) a neural network.

decided, the function computed by the neural network is decided. There are many neural network learning algorithms. The most commonly applied idea for training a multi-layer feed-forward neural network is that, as long as the activation function is differentiable, the whole neural network can be regarded as a differentiable function which can be optimized by **gradient descent** method.

The most successful algorithm, **Back-Propagation (BP)** [Werbos, 1974, Rumelhart et al., 1986], works as follows. At first, the inputs are feed-forwarded from the input layer via the hidden layer to the output layer, at which the error is calculated by comparing the network output with the ground-truth. Then, the error will be back propagated to the hidden layer and the input layer, during which the connection weights and biases are adjusted to reduce the error. The process is accomplished by tuning towards the direction with the gradient. Such a process will be repeated in many rounds, until the training error is minimized or the training process is terminated to avoid overfitting.

1.2.4 Naïve Bayes Classifier

To classify a test instance $\boldsymbol{x}$, one approach is to formulate a probabilistic model to estimate the posterior probability $P(y \mid \boldsymbol{x})$ of different y's, and predict the one with the largest posterior probability; this is the ***maximum a posterior* (MAP)** rule. By *Bayes Theorem*, we have

$$P(y \mid \boldsymbol{x}) = \frac{P(\boldsymbol{x} \mid y)P(y)}{P(\boldsymbol{x})}\,, \tag{1.14}$$

where $P(y)$ can be estimated by counting the proportion of class y in the training set, and $P(\boldsymbol{x})$ can be ignored since we are comparing different y's on the same $\boldsymbol{x}$. Thus we only need to consider $P(\boldsymbol{x} \mid y)$. If we can get an accurate estimate of $P(\boldsymbol{x} \mid y)$, we will get the best classifier in theory from the given training data, that is, the **Bayes optimal classifier** with the **Bayes error rate**, the smallest error rate in theory. However, estimating $P(\boldsymbol{x} \mid y)$ is not straightforward, since it involves the estimation of exponential numbers of joint-probabilities of the features. To make the estimation tractable, some assumptions are needed.

The naïve Bayes classifier assumes that, given the class label, the n features are independent of each other within each class. Thus, we have

$$P(\boldsymbol{x} \mid y) = \prod_{i=1}^{n} P(x_i \mid y), \tag{1.15}$$

which implies that we only need to estimate each feature value in each class in order to estimate the conditional probability, and therefore the calculation of joint-probabilities is avoided.

In the training stage, the naïve Bayes classifier estimates the probabilities $P(y)$ for all classes $y \in \mathcal{Y}$, and $P(x_i \mid y)$ for all features $i = 1, \ldots, n$ and all feature values x_i from the training set. In the test stage, a test instance $\boldsymbol{x}$ will be predicted with label y if y leads to the largest value of

$$P(y \mid \boldsymbol{x}) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y) \tag{1.16}$$

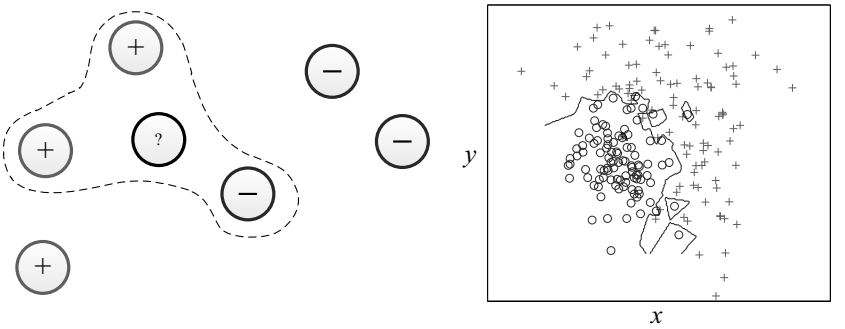
among all the class labels.

1.2.5 k-Nearest Neighbor

The k-nearest neighbor (kNN) algorithm relies on the principle that objects similar in the input space are also similar in the output space. It is a **lazy learning** approach since it does not have an explicit training process, but simply stores the training set instead. For a test instance, a k-nearest neighbor learner identifies the k instances from the training set that are closest to the test instance. Then, for classification, the test instance will be classified to the majority class among the k instances; while for regression, the test instance will be assigned the average value of the k instances. Figure 1.6(a) illustrates how to classify an instance by a 3-nearest neighbor classifier. Figure 1.6(b) shows the decision boundary of a 1-nearest neighbor classifier, also called the **nearest neighbor classifier**.

1.2.6 Support Vector Machines and Kernel Methods

Support vector machines (SVMs) [Cristianini and Shawe-Taylor, 2000], originally designed for binary classification, are **large margin classifiers**



(a) (b)

FIGURE 1.6: Illustration of (a) how a k-nearest neighbor classifier predicts on a test instance, and (b) the decision boundary of the nearest neighbor classifier on the *three-Gaussians* data set.

that try to separate instances of different classes with the maximum margin hyperplane. The **margin** is defined as the minimum distance from instances of different classes to the classification hyperplane.

Considering a linear classifier $y = \texttt{sign}(\boldsymbol{w}^\top \boldsymbol{x}+b)$, or abbreviated as $(\boldsymbol{w}, b)$, we can use the **hinge loss** to evaluate the fitness to the data:

$$\sum_{i=1}^{m} \max\{0, 1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)\}. \tag{1.17}$$

The Euclidean distance from an instance $\boldsymbol{x}_i$ to the hyperplane $\boldsymbol{w}^\top \boldsymbol{x} + b$ is

$$\frac{|\boldsymbol{w}^\top \boldsymbol{x}_i + b|}{\|\boldsymbol{w}\|}. \tag{1.18}$$

If we restrict $|\boldsymbol{w}^\top \boldsymbol{x}_i + b| \geq 1$ for all instances, the minimum distance to the hyperplane is $\|\boldsymbol{w}\|^{-1}$. Therefore, SVMs maximize $\|\boldsymbol{w}\|^{-1}$.

Thus, SVMs solve the optimization problem

$$\begin{aligned} (\boldsymbol{w}^*, b^*) = \arg\min_{\boldsymbol{w}, b, \xi_i} \frac{\|\boldsymbol{w}\|^2}{2} + C\sum_{i=1}^{m} \xi_i \\ s.t. \quad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i \ (\forall i = 1, \ldots, m) \\ \xi_i \geq 0 \ (\forall i = 1, \ldots, m)\,, \end{aligned} \tag{1.19}$$

where C is a parameter and ξ_i's are *slack variables* introduced to enable the learner to deal with data that could not be perfectly separated, such as data with noise. An illustration of an SVM is shown in Figure 1.7.

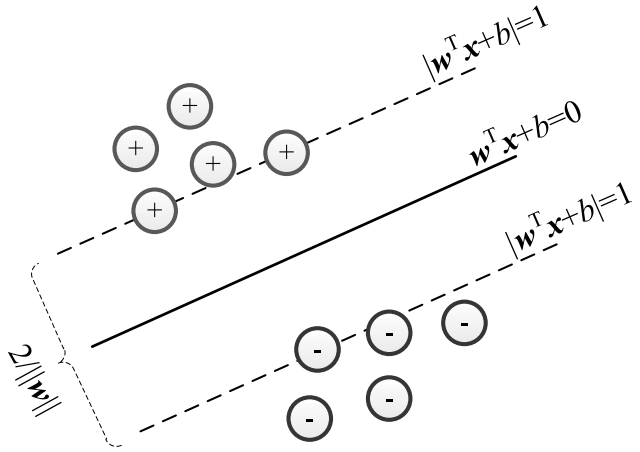


FIGURE 1.7: Illustration of SVM.

(1.19) is called the *primal* form of the optimization. The *dual* form, which gives the same optimal solution, is

$$\begin{aligned} \boldsymbol{\alpha}^* = \arg\max_{\boldsymbol{\alpha}} & \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \\ s.t. \quad & \sum_{i=1}^{m} \alpha_i y_i = 0 \\ & \alpha_i \geq 0 \, (\forall i = 1, \ldots, m) \, , \end{aligned} \tag{1.20}$$

where $\langle \cdot, \cdot \rangle$ is the inner product. The solution $\boldsymbol{w}^*$ of the primal form is now presented as

$$\boldsymbol{w}^* = \sum_{i=1}^{m} \alpha_i^* y_i \boldsymbol{x}_i \, , \tag{1.21}$$

and the inner product between $\boldsymbol{w}^*$ and an instance $\boldsymbol{x}$ can be calculated as

$$\langle \boldsymbol{w}^*, \boldsymbol{x} \rangle = \sum_{i=1}^{m} \alpha_i^* y_i \langle \boldsymbol{x}_i, \boldsymbol{x} \rangle \, . \tag{1.22}$$

A limitation of the linear classifiers is that, when the data is intrinsically nonlinear, linear classifiers cannot separate the classes well. In such cases, a general approach is to map the data points onto a higher-dimensional feature space where the data linearly non-separable in the original feature space become linearly separable. However, the learning process may become very slow and even intractable since the inner product will be difficult to calculate in the high-dimensional space.

Fortunately, there is a class of functions, **kernel functions** (also called **kernels**), which can help address the problem. The feature space derived by kernel functions is called the **Reproducing Kernel Hilbert Space** (**RKHS**). An inner product in the RKHS equals kernel mapping of inner product of instances in the original lower-dimensional feature space. In other words,

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle \tag{1.23}$$

for all $\boldsymbol{x}_i$'s, where ϕ is a mapping from the original feature space to a higher-dimensional space and K is a kernel. Thus, we can simply replace the inner products in the dual form of the optimization by the kernel.

According to *Mercer's Theorem* [Cristianini and Shawe-Taylor, 2000], every positive semi-definite symmetric function is a kernel. Popular kernels include the **linear kernel**

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \,, \tag{1.24}$$

the **polynomial kernel**

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle^d \,, \tag{1.25}$$

where d is the degree of the polynomial, and the **Gaussian kernel** (or called **RBF kernel**)

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right) \,, \tag{1.26}$$

where σ is the parameter of the Gaussian width.

The **kernel trick**, i.e., mapping the data points with a kernel and then accomplishing the learning task in the RKHS, is a general strategy that can be incorporated into any learning algorithm that considers only inner products between the input feature vectors. Once the kernel trick is used, the learning algorithms are called **kernel methods**. Indeed, SVMs are a special kind of kernel method, i.e., linear classifiers facilitated with kernel trick.

1.3 Evaluation and Comparison

Usually, we have multiple alternative learning algorithms to choose among, and a number of parameters to tune. The task of choosing the best algorithm and the settings of its parameters is known as **model selection**, and for this purpose we need to estimate the performance of the learner. By empirical ways, this involves design of experiments and statistical hypothesis tests for comparing the models.

It is unwise to estimate the generalization error of a learner by its **training error**, i.e., the error that the learner makes on the training data, since

training error prefers complex learners rather than learners that generalize well. Usually, a learner with very high complexity can have zero training error, such as a fully grown decision tree; however, it is likely to perform badly on unseen data due to overfitting. A proper process is to evaluate the performance on a validation set. Note that the labels in the training set and validation set are known *a priori* to the training process, and should be used together to derive and tune the final learner once the model has been selected.

In fact, in most cases the training and validation sets are obtained by splitting a given data set into two parts. While splitting, the properties of the original data set should be kept as much as possible; otherwise the validation set may provide misleading estimates, for an extreme example, the training set might contain only positive instances while the validation set contains only negative instances. In classification, when the original data set is split randomly, the class percentage should be maintained for both training and validation sets; this is called **stratification**, or *stratified sampling*.

When there is not enough labeled data available to create a separate validation set, a commonly used validation method is **cross-validation**. In k**-fold cross-validation**, the original data set is partitioned by stratified split into k equal-size disjoint subsets, $D_1, \ldots, D_k$, and then k runs of training-tests are performed. In the ith run, D_i is used as the validation set while the union of all the other subsets, i.e., $\bigcup_{j \neq i} D_j$, is used as the training set. The average results of the k runs are taken as the results of the cross-validation. To reduce the influence of randomness introduced by data split, the k-fold cross-validation can be repeated t times, which is called t**-times** k**-fold cross-validation**. Usual configurations include *10-times 10-fold cross-validation*, and *5-times 2-fold cross-validation* suggested by Dietterich [1998]. Extremely, when k equals the number of instances in the original data set, there is only one instance in each validation set; this is called **leave-one-out (LOO)** validation.

After obtaining the estimated errors, we can compare different learning algorithms. A simple comparison on average errors, however, is not reliable since the winning algorithm may occasionally perform well due to the randomness in data split. **Hypothesis test** is usually employed for this purpose.

To compare learning algorithms that are efficient enough to run 10 times, the $\mathbf{5 \times 2}$ **cv paired** t**-test** is a good choice [Dietterich, 1998]. In this test, we run 5-times 2-fold cross-validation. In each run of 2-fold cross-validation, the data set D is randomly split into two subsets D_1 and D_2 of equal size. Two algorithms a and b are trained on each set and tested on the other, resulting in four error estimates: $err_a^{(1)}$ and $err_b^{(1)}$ (trained on D_1 and tested on D_2) and $err_a^{(2)}$ and $err_b^{(2)}$ (trained on D_2 and tested on D_1). We have the error differences

$$d^{(i)} = err_a^{(i)} - err_b^{(i)} \qquad (i = 1, 2) \tag{1.27}$$

with the mean and the variance, respectively:

$$\mu = \frac{d^{(1)} + d^{(2)}}{2} , \tag{1.28}$$

$$s^2 = (d^{(1)} - \mu)^2 + (d^{(2)} - \mu)^2 . \tag{1.29}$$

Let s_i^2 denote the variance in the ith time 2-fold cross-validation, and $d_1^{(1)}$ denote the error difference in the first time. Under the null hypothesis, the 5×2 cv $\tilde{t}$-statistic

$$\tilde{t} = \frac{d_1^{(1)}}{\sqrt{\frac{1}{5}\sum_{i=1}^{5} s_i^2}} \sim t_5 , \tag{1.30}$$

would be distributed according to the *Student's t-distribution* with 5 degrees of freedom. We then choose a significance level α. If $\tilde{t}$ falls into the interval $[-t_5(\alpha/2), t_5(\alpha/2)]$, the null hypothesis is accepted, suggesting that there is no significant difference between the two algorithms. Usually α is set to 0.05 or 0.1.

To compare learning algorithms that can be run only once, the **McNemar's test** can be used instead [Dietterich, 1998]. Let err_{01} denote the number of instances on which the first algorithm makes a wrong prediction while the second algorithm is correct, and err_{10} denotes the inverse. If the two algorithms have the same performance, err_{01} is close to err_{10}, and therefore, the quantity

$$\frac{(|err_{01} - err_{10}| - 1)^2}{err_{01} + err_{10}} \sim \chi_1^2 \tag{1.31}$$

would be distributed according to the *χ^2-distribution*.

Sometimes, we evaluate multiple learning algorithms on multiple data sets. In this situation, we can conduct the **Friedman test** [Demšar, 2006]. First, we sort the algorithms on each data set according to their average errors. On each data set, the best algorithm is assigned rank 1, the worse algorithms are assigned increased ranks, and average ranks are assigned in case of ties. Then, we average the ranks of each algorithm over all data sets, and use the *Nemenyi post-hoc test* [Demšar, 2006] to calculate the *critical difference* value

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} , \tag{1.32}$$

where k is the number of algorithms, N is the number of data sets and q_α is the *critical value* [Demšar, 2006]. A pair of algorithms are believed to be significantly different if the difference of their average ranks is larger than the critical difference.

The Friedman test results can be visualized by plotting the **critical difference diagram**, as illustrated in Figure 1.8, where each algorithm corresponds to a bar centered at the average rank with the width of critical difference value. Figure 1.8 discloses that the algorithm A is significantly better

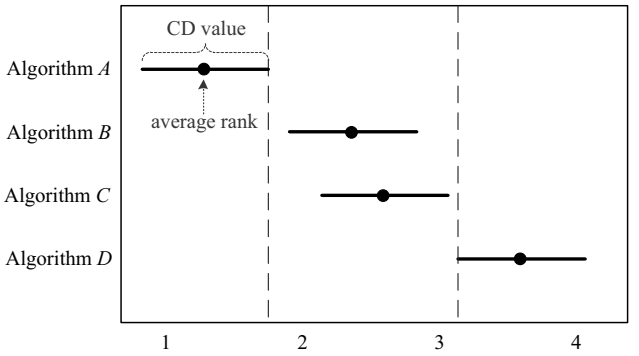


FIGURE 1.8: Illustration of critical difference diagram.

than all the other algorithms, the algorithm D is significantly worse than all the other algorithms, and the algorithms B and C are not significantly different, according to the given significance level.

1.4 Ensemble Methods

Ensemble methods train multiple learners to solve the same problem. In contrast to ordinary learning approaches which try to construct one learner from training data, ensemble methods try to construct a set of learners and combine them. Ensemble learning is also called **committee-based learning**, or learning **multiple classifier systems**.

Figure 1.9 shows a common ensemble architecture. An ensemble contains a number of learners called **base learners**. Base learners are usually generated from training data by a **base learning algorithm** which can be decision tree, neural network or other kinds of learning algorithms. Most ensemble methods use a single base learning algorithm to produce *homogeneous* base learners, i.e., learners of the same type, leading to **homogeneous ensembles**, but there are also some methods which use multiple learning algorithms to produce *heterogeneous* learners, i.e., learners of different types, leading to **heterogeneous ensembles**. In the latter case there is no single base learning algorithm and thus, some people prefer calling the learners **individual learners** or **component learners** to *base learners*.

The generalization ability of an ensemble is often much stronger than that of base learners. Actually, ensemble methods are appealing mainly because they are able to boost **weak learners** which are even just slightly better than random guess to **strong learners** which can make very accurate predictions. So, *base learners* are also referred to as *weak learners*.

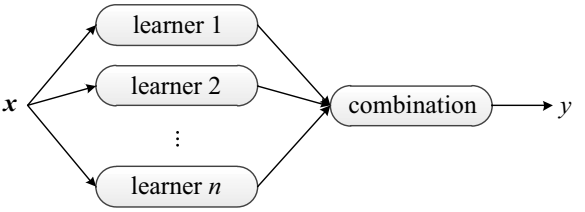


FIGURE 1.9: A common ensemble architecture.

It is difficult to trace the starting point of the history of ensemble methods since the basic idea of deploying multiple models has been in use in human society for a long time. For example, even earlier than the introduction of **Occam's razor**, the common basic assumption of scientific research which prefers simple hypotheses to complex ones when both fit empirical observations well, the Greek philosopher Epicurus (341 - 270 B.C.) introduced the **principle of multiple explanations** [Asmis, 1984] which advocated to keep all hypotheses that are consistent with empirical observations.

There are three threads of early contributions that led to the current area of ensemble methods; that is, **combining classifiers**, **ensembles of weak learners** and **mixture of experts**. *Combining classifiers* was mostly studied in the pattern recognition community. In this thread, researchers generally work on strong classifiers, and try to design powerful *combining rules* to get stronger combined classifiers. As the consequence, this thread of work has accumulated deep understanding on the design and use of different combining rules. *Ensembles of weak learners* was mostly studied in the machine learning community. In this thread, researchers often work on weak learners and try to design powerful algorithms to boost the performance from weak to strong. This thread of work has led to the birth of famous ensemble methods such as AdaBoost, Bagging, etc., and theoretical understanding on why and how weak learners can be boosted to strong ones. *Mixture of experts* was mostly studied in the neural networks community. In this thread, researchers generally consider a **divide-and-conquer** strategy, try to learn a mixture of parametric models jointly and use combining rules to get an overall solution.

Ensemble methods have become a major learning paradigm since the 1990s, with great promotion by two pieces of pioneering work. One is empirical [Hansen and Salamon, 1990], in which it was found that predictions made by the combination of a set of classifiers are often more accurate than predictions made by the best single classifier. A simplified illustration is shown in Figure 1.10. The other is theoretical [Schapire, 1990], in which it was proved that weak learners can be boosted to strong learners. Since strong learners are desirable yet difficult to get, while weak learners are easy to obtain in real practice, this result opens a promising direction of gener-

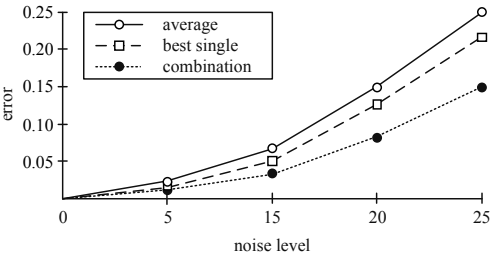


FIGURE 1.10: A simplified illustration of Hansen and Salamon [1990]'s observation: Ensemble is often better than the best single.

ating strong learners by ensemble methods.

Generally, an ensemble is constructed in two steps, i.e., generating the base learners, and then combining them. To get a good ensemble, it is generally believed that the base learners should be as *accurate* as possible, and as *diverse* as possible.

It is worth mentioning that generally, the computational cost of constructing an ensemble is not much larger than creating a single learner. This is because when we want to use a single learner, we usually need to generate multiple versions of the learner for model selection or parameter tuning; this is comparable to generating base learners in ensembles, while the computational cost for combining base learners is often small since most combination strategies are simple.

1.5 Applications of Ensemble Methods

The *KDD-Cup* [1] is the most famous data mining competition. Since 1997, it is held every year and attracts the interests of data mining teams all over the world. The competition problems cover a large variety of practical tasks, such as network intrusion detection (1999), molecular bioactivity & protein locale prediction (2001), pulmonary embolisms detection (2006), customer relationship management (2009), educational data mining (2010), music recommendation (2011), etc. In the past KDD-Cup competitions, among various techniques utilized in the solutions, ensemble methods have drawn the most attention and won the competitions for the most times. For example, in KDD-Cups of the last three years (2009-2011), all the first-place and second-place winners used ensemble methods.

[1] `http://www.sigkdd.org/kddcup/`.

Another famous competition, the *Netflix Prize*,[2] is held by the online DVD-rental service Netflix and seeks to improve the accuracy of predictions about how much someone is going to enjoy a movie based on their preferences; if one participating team improves Netflix's own algorithm by 10% accuracy, they would win the grand prize of $1,000,000. On September 21, 2009, Nexflix awarded the $1M grand prize to the team *BellKor's Pragmatic Chaos*, whose solution was based on combining various classifiers including asymmetric factor models, regression models, restricted Boltzmann machines, matrix factorization, k-nearest neighbor, etc. Another team, which achieved the winning performance but was defeated because the result was submitted 20 minutes later, even used *The Ensemble* as the team name.

In addition to the impressive results in competitions, ensemble methods have been successfully applied to diverse real-world tasks. Indeed, they have been found useful in almost all places where learning techniques are exploited. For example, computer vision has benefited much from ensemble methods in almost all branches such as object detection, recognition and tracking.

Viola and Jones [2001, 2004] proposed a general object detection framework by combining AdaBoost with a cascade architecture. Viola and Jones [2004] reported that, on a 466MHz machine, the face detector spent only 0.067 seconds for a 384×288 image; this is almost 15 times faster than state-of-the-art face detectors, while the detection accuracy is comparable. This framework was recognized as one of the most exciting breakthroughs in computer vision (especially, face detection) during the past decade.

Huang et al. [2000] designed an ensemble architecture for pose-invariant face recognition, particularly for recognizing faces with in-depth rotations. The basic idea is to combine a number of view-specific neural networks with a specially designed combination module. In contrast to conventional techniques which require pose information as input, this framework does not need pose information and it can even output pose estimation in addition to the recognition result. Huang et al. [2000] reported that this framework even outperformed conventional techniques facilitated with perfect pose information. A similar method was later applied to multi-view face detection [Li et al., 2001].

Object tracking aims to assign consistent labels to the target objects in consecutive frames of a video. By considering tracking as a binary classification problem, Avidan [2007] proposed *ensemble tracking*, which trains an ensemble online to distinguish between the object and the background. This framework constantly updates a set of weak classifiers, which can be added or removed at any time to incorporate new information about changes in object appearance and the background. Avidan [2007] showed

[2] http://www.netflixprize.com/.

that the ensemble tracking framework could work in a large variety of videos with various object size, and it runs very efficiently, at a few frames per second without optimization, hence can be used in online applications.

Ensemble methods have been found very appropriate to characterize computer security problems because each activity performed on computer systems can be observed at *multiple abstraction levels*, and the relevant information may be collected from *multiple information sources* [Corona et al., 2009].

Giacinto et al. [2003] applied ensemble methods to intrusion detection. Considering that there are different types of features characterizing the connection, they constructed an ensemble from each type of features independently, and then combined the outputs from these ensembles to produce the final decision. Giacinto et al. [2003] reported that, when detecting known attacks, ensemble methods lead to the best performance. Later, Giacinto et al. [2008] proposed an ensemble method for anomaly-based intrusion detection which is able to detect intrusions never seen before.

Malicious executables are programs designed to perform a malicious function without the owner's permission, and they generally fall into three categories, i.e., viruses, worms, and Trojan horses. Schultz et al. [2001] proposed an ensemble method to detect previously unseen malicious executables automatically, based on representing the programs using binary profiling, string sequences and hex dumps. Kolter and Maloof [2006] represented programs using n-grams of byte codes, and reported that boosted decision trees achieved the best performance; they also suggested that this method could be used as the basis for an operational system for detecting new malicious executables never seen before.

Ensemble methods have been found very useful in diverse tasks of computer aided medical diagnosis, particularly for increasing the diagnosis reliability.

Zhou et al. [2002a] designed a two-layered ensemble architecture for lung cancer cell identification, where the first layer predicts benign cases if and only if all component learners agree, and otherwise the case will be passed to the second layer to make a further decision among benign and different cancer types. Zhou et al. [2002a] reported that the two-layered ensemble results in a high identification rate with a low false-negative identification rate.

For early diagnosis of Alzheimer's disease, previous methods generally considered single channel data from the EEG (electroencephalogram). To make use of multiple data channels, Polikar et al. [2008] proposed an ensemble method where the component learners are trained on different data sources obtained from different electrodes in response to different stimuli and in different frequency bands, and their outputs are combined for the final diagnosis.

In addition to computer vision, computer security and computer aided medical diagnosis, ensemble methods have also been applied to many

other domains and tasks such as credit card fraud detection [Chan et al., 1999, Panigrahi et al., 2009], bankruptcy prediction [West et al., 2005], protein structure classification [Tan et al., 2003, Shen and Chou, 2006], species distributions forecasting [Araújo and New, 2007], weather forecasting [Maqsood et al., 2004, Gneiting and Raftery, 2005], electric load forecasting [Taylor and Buizza, 2002], aircraft engine fault diagnosis [Goebel et al., 2000, Yan and Xue, 2008], musical genre and artist classification [Bergstra et al., 2006], etc.

1.6 Further Readings

There are good textbooks on machine learning [Mitchell, 1997, Alpaydin, 2010, Bishop, 2006, Hastie et al., 2001], pattern recognition [Duda et al., 2000, Theodoridis and Koutroumbas, 2009, Ripley, 1996, Bishop, 1995] and data mining [Han and Kamber, 2006, Tan et al., 2006, Hand et al., 2001]. More introductory materials can be found in these books.

Linear discriminant analysis is closely related to **principal component analysis** (PCA) [Jolliffe, 2002], both looking for linear combination of features to represent the data. LDA is a supervised approach focusing on distinguishing between different classes, while PCA is an unsupervised approach generally used to identify the largest variability. Decision trees can be mapped to a set of "if-then" rules [Quinlan, 1993]. Most decision trees use splits like "$x \geq 1$" or "$y \geq 2$", leading to axis-parallel partitions of instance space. There are also exceptions, e.g., **oblique decision trees** [Murthy et al., 1994] which use splits like "$x+y \geq 3$", leading to non-axis-parallel partitions. The BP algorithm is the most popular and most successful neural network learning algorithm. It has many variants, and can also be used to train neural networks whose structures are different from feed-forward networks, such as **recurrent neural networks** where there are cross-layer connections. Haykin [1998] provides a good introduction to neural networks. Though the nearest neighbor algorithm is very simple, it works well in most cases. The error of the nearest neighbor classifier is guaranteed to be no worse than twice of the Bayes error rate on infinite data [Cover and Hart, 1967], and kNN approaches the Bayes error rate for some k value which is related to the amount of data. The distances between instances are not constrained to be calculated by the Euclidean distance, and the contributions from different neighbors can be weighted. More information on kNN can be found in [Dasarathy, 1991]. The naïve Bayes classifier based on the conditional independence assumption works well in most cases [Domingos and Pazzani, 1997]; however, it is believed that the performance can be improved further by relaxing the assumption, and therefore

many **semi-naïve Bayes classifiers** such as TAN [Friedman et al., 1997] and LBR [Zheng and Webb, 2000] have been developed. A particularly successful one is the AODE [Webb et al., 2005], which has incorporated ensemble mechanism and often beats TAN and LBR, especially on intermediate-size data sets. SVMs are rooted in the **statistical learning theory** [Vapnik, 1998]. More introductory materials on SVMs and kernel methods can be found in [Cristianini and Shawe-Taylor, 2000, Schölkopf et al., 1999].

Introductory materials on hypothesis tests can be found in [Fleiss, 1981]. Different hypothesis tests are usually based on different assumptions, and should be applied in different situations. The 10-fold cross-validation t-test was popularly used; however, Dietterich [1998] discloses that such a test underestimates the variability and it is likely to incorrectly detect a difference when no difference exists (i.e., the type I error), while the 5×2cv paired t-test is recommended instead.

The **No Free Lunch Theorem** [Wolpert, 1996, Wolpert and Macready, 1997] implies that it is hopeless to dream for a learning algorithm which is consistently better than other learning algorithms. It is important to notice, however, that the No Free Lunch Theorem considers the whole problem space, that is, all the possible learning tasks; while in real practice, we are usually only interested in a give task, and in such a situation, the effort of trying to find the best algorithm is valid. From the experience of the author of this book, for lots of tasks, the best off-the-shelf learning technique at present is ensemble methods such as Random Forest facilitated with **feature engineering** which constructs/generates usually an overly large number of new features rather than simply working on the original features.

[Kuncheva, 2004] and [Rokach, 2010] are books on ensemble methods. Xu and Amari [2009] discuss the relation between *combining classifiers* and *mixture of experts*. The *MCS* workshop (*International Workshop on Multiple Classifier Systems*) is the major forum in this area. Abundant literature on ensemble methods can also be found in various journals and conferences on machine learning, pattern recognition and data mining.

This page intentionally left blank

References

N. Abe and H. Mamitsuka. Query learning strategies using Boosting and Bagging. In *Proceedings of the 15th International Conference on Machine Learning*, pages 1–9, Madison, WI, 1998.

R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 94–105, Seattle, WA, 1998.

M. R. Ahmadzadeh and M. Petrou. Use of Dempster-Shafer theory to combine classifiers which use different class boundaries. *Pattern Analysis and Application*, 6(1):41–46, 2003.

A. Al-Ani and M. Deriche. A new technique for combining multiple classifiers using the Dempster-Shafer theory of evidence. *Journal of Artificial Intelligence Research*, 17(1):333–361, 2002.

K. M. Ali and M. J. Pazzani. Error reduction through learning multiple descriptions. *Machine Learning*, 24(3):173–202, 1996.

E. L. Allwein, R. E. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141, 2000.

E. Alpaydin. *Introduction to Machine Learning*. MIT Press, Cambridge, MA, 2nd edition, 2010.

M. R. Anderberg. *Cluster Analysis for Applications*. Academic, New York, NY, 1973.

R. Andrews, J. Diederich, and A. B. Tickle. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Systems*, 8(6):373–389, 1995.

M. Ankerst, M. Breunig, H.-P. Kriegel, and J. Sander. OPTICS: Ordering points to identify the clustering structure. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 49–60, Philadelphia, PA, 1999.

M. Anthony and N. Biggs. *Computational Learning Theory*. Cambridge University Press, Cambridge, UK, 1992.

M. B. Araújo and M. New. Ensemble forecasting of species distributions. *Trends in Ecology & Evolution*, 22(1):42–47, 2007.

J. A. Aslam and S. E. Decatur. General bounds on statistical query learning and PAC learning with noise via hypothesis boosting. In *Proceedings of the 35th IEEE Annual Symposium on Foundations of Computer Science*, pages 282–291, Palo Alto, CA, 1993.

E. Asmis. *Epicurus' Scientific Method*. Cornell University Press, Ithaca, NY, 1984.

A. V. Assche and H. Blockeel. Seeing the forest through the trees: Learning a comprehensible model from an ensemble. In *Proceedings of the 18th European Conference on Machine Learning*, pages 418–429, Warsaw, Poland, 2007.

S. Avidan. Ensemble tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):261–271, 2007.

R. Avogadri and G. Valentini. Fuzzy ensemble clustering based on random projections for DNA microarray data analysis. *Artificial Intelligence in Medicine*, 45(2-3):173–183, 2009.

H. Ayad and M. Kamel. Finding natural clusters using multi-clusterer combiner based on shared nearest neighbors. In *Proceedings of the 4th International Workshop on Multiple Classifier Systems*, pages 166–175, Surrey, UK, 2003.

F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004.

B. Bakker and T. Heskes. Clustering ensembles of neural network models. *Neural Networks*, 16(2):261–269, 2003.

M.-F. Balcan, A. Z. Broder, and T. Zhang. Margin based active learning. In *Proceedings of the 20th Annual Conference on Learning Theory*, pages 35–50, San Diego, CA, 2007.

R. E. Banfield, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer. Ensemble diversity measures and their application to thinning. *Information Fusion*, 6(1):49–62, 2005.

E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1-2): 105–139, 1999.

K. Bennett, A. Demiriz, and R. Maclin. Exploiting unlabeled data in ensemble methods. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 289–296, Edmonton, Canada, 2002.

J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kégl. Aggregate features and AdaBoost for music classification. *Machine Learning*, 65(2-3):473–484, 2006.

Y. Bi, J. Guan, and D. Bell. The combination of multiple classifiers using an evidential reasoning approach. *Artificial Intelligence*, 172(15):1731–1751, 2008.

P. J. Bickel, Y. Ritov, and A. Zakai. Some theory for generalized boosting algorithms. *Journal of Machine Learning Research*, 7:705–732, 2006.

J. A. Bilmes. A gentle tutorial of the EM algorithm and its applications to parameter estimation for Gaussian mixture and hidden Markov models. Technical Report TR-97-021, Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA, 1998.

C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, New York, NY, 1995.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, NY, 2006.

C. M. Bishop and M. Svensén. Bayesian hierarchical mixtures of experts. In *Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence*, pages 57–64, Acapulco, Mexico, 2003.

A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100, Madison, WI, 1998.

J. K. Bradley and R. E. Schapire. FilterBoost: Regression and classification on large datasets. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 185–192. MIT Press, Cambridge, MA, 2008.

U. Brefeld and T. Scheffer. AUC maximizing support vector learning. In *Proceedings of the ICML 2005 Workshop on ROC Analysis in Machine Learning*, Bonn, Germany, 2005.

U. Brefeld, P. Geibel, and F. Wysotzki. Support vector machines with example dependent costs. In *Proceedings of the 14th European Conference on Machine Learning*, pages 23–34, Cavtat-Dubrovnik, Croatia, 2003.

L. Breiman. Bias, variance, and arcing classifiers. Technical Report 460, Statistics Department, University of California, Berkeley, CA, 1996a.

L. Breiman. Stacked regressions. *Machine Learning*, 24(1):49–64, 1996b.

L. Breiman. Out-of-bag estimation. Technical report, Department of Statistics, University of California, 1996c.

L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996d.

L. Breiman. Prediction games and arcing algorithms. *Neural Computation*, 11(7):1493–1517, 1999.

L. Breiman. Randomizing outputs to increase prediction accuracy. *Machine Learning*, 40(3):113–120, 2000.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

L. Breiman. Population theory for boosting ensembles. *Annals of Statistics*, 32(1):1–11, 2004.

L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and Regression Trees*. Chapman and Hall/CRC, Boca Raton, FL, 1984.

G. Brown. An information theoretic perspective on multiple classifier systems. In *Proceedings of the 8th International Workshop on Multiple Classifier Systems*, pages 344–353, Reykjavik, Iceland, 2009.

G. Brown. Some thoughts at the interface of ensemble methods and feature selection. Keynote at the 9th International Workshop on Multiple Classifier Systems, Cairo, Egypt, 2010.

G. Brown, J. L. Wyatt, R. Harris, and X. Yao. Diversity creation methods: A survey and categorisation. *Information Fusion*, 6(1):5–20, 2005a.

G. Brown, J. L. Wyatt, and P. Tino. Managing diversity in regression ensembles. *Journal of Machine Learning Research*, 6:1621–1650, 2005b.

P. Bühlmann and B. Yu. Analyzing bagging. *Annals of Statistics*, 30(4):927–961, 2002.

P. Bühlmann and B. Yu. Boosting with the l_2 loss: Regression and classification. *Journal of the American Statistical Association*, 98(462):324–339, 2003.

A. Buja and W. Stuetzle. The effect of bagging on variance, bias, and mean squared error. Technical report, AT&T Labs-Research, 2000a.

A. Buja and W. Stuetzle. Smoothing effects of bagging. Technical report, AT&T Labs-Research, 2000b.

A. Buja and W. Stuetzle. Observations on bagging. *Statistica Sinica*, 16(2): 323–351, 2006.

R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes. Ensemble selection from libraries of models. In *Proceedings of the 21st International Conference on Machine Learning*, pages 18–23, Banff, Canada, 2004.

P. D. Castro, G. P. Coelho, M. F. Caetano, and F. J. V. Zuben. Designing ensembles of fuzzy classification systems: An immune-inspired approach.

In *Proceedings of the 4th International Conference on Artificial Immune Systems*, pages 469–482, Banff, Canada, 2005.

P. Chan and S. Stolfo. Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. In *Proceeding of the 4th International Conference on Knowledge Discovery and Data Mining*, pages 164–168, New York, NY, 1998.

P. K. Chan, W. Fan, A. L. Prodromidis, and S. J. Stolfo. Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems*, 14(6):67–74, 1999.

V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):1–58, 2009.

O. Chapelle and A. Zien. Semi-supervised learning by low density separation. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, pages 57–64. Barbados, 2005.

O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

N. V. Chawla. Data mining for imbalanced datasets: An overview. In O. Maimon and L. Rokach, editors, *The Data Mining and Knowledge Discovery Handbook*, pages 853–867. Springer, New York, NY, 2006.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority oversampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer. SMOTEBoost: Improving prediction of the minority class in boosting. In *Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 107–119, Cavtat-Dubrovnik, Croatia, 2003.

H. Chen, P. Tiňo, and X. Yao. A probabilistic ensemble pruning algorithm. In *Working Notes of ICDM'06 Workshop on Optimization-Based Data Mining Techniques with Applications*, pages 878–882, Hong Kong, China, 2006.

H. Chen, P. Tiňo, and X. Yao. Predictive ensemble pruning by expectation propagation. *IEEE Transactions on Knowledge and Data Engineering*, 21 (7):999–1013, 2009.

B. Clarke. Comparing Bayes model averaging and stacking when model approximation error cannot be ignored. *Journal of Machine Learning Research*, 4:683–712, 2003.

A. L. V. Coelho, C. A. M. Lima, and F. J. V. Zuben. GA-based selection of components for heterogeneous ensembles of support vector machines.

In *Proceedings of the Congress on Evolutionary Computation*, pages 2238–2244, Canberra, Australia, 2003.

J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

I. Corona, G. Giacinto, C. Mazzariello, F. Roli, and C. Sansone. Information fusion for computer security: State of the art and open issues. *Information Fusion*, 10(4):274–284, 2009.

T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, New York, NY, 1991.

M. Coyle and B. Smyth. On the use of selective ensembles for relevance classification in case-based web search. In *Proceedings of the 8th European Conference on Case-Based Reasoning*, pages 370–384, Fethiye, Turkey, 2006.

K. Crammer and Y. Singer. On the learnability and design of output codes for multiclass problems. *Machine Learning*, 47(2-3):201–233, 2002.

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, Cambridge, UK, 2000.

P. Cunningham and J. Carney. Diversity versus quality in classification ensembles based on feature selection. Technical Report TCD-CS-2000-02, Department of Computer Science, Trinity College Dublin, 2000.

A. Cutler and G. Zhao. PERT - perfect random tree ensembles. In *Proceedings of the 33rd Symposium on the Interface of Computing Science and Statistics*, pages 490–497, Costa Mesa, CA, 2001.

I. Dagan and S. P. Engelson. Committee-based sampling for training probabilistic classifiers. In *Proceedings of the 12th International Conference on Machine Learning*, pages 150–157, San Francisco, CA, 1995.

F. d'Alché-Buc, Y. Grandvalet, and C. Ambroise. Semi-supervised marginboost. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 553–560. MIT Press, Cambridge, MA, 2002.

B. V. Dasarathy, editor. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, Los Alamitos, CA, 1991.

S. Dasgupta. Analysis of a greedy active learning strategy. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 337–344. MIT Press, Cambridge, MA, 2005.

S. Dasgupta. Coarse sample complexity bounds for active learning. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 235–242. MIT Press, Cambridge, MA, 2006.

S. Dasgupta and D. Hsu. Hierarchical sampling for active learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 208–215, Helsinki, Finland, 2008.

S. Dasgupta, A. T. Kalai, and C. Monteleoni. Analysis of perceptron-based active learning. In *Proceedings of the 18th Annual Conference on Learning Theory*, pages 249–263, Bertinoro, Italy, 2005.

J. Davis and M. Goadrich. The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 233–240, Pittsburgh, PA, 2006.

W. H. E. Day and H. Edelsbrunner. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of Classification*, 1:7–24, 1984.

N. C. de Concorcet. *Essai sur l'Application de l'Analyze à la Probabilité des Décisions Rendues à la Pluralité des Voix*. Imprimérie Royale, Paris, France, 1785.

A. Demiriz, K. P. Bennett, and J. Shawe-Taylor. Linear programming boosting via column generation. *Machine Learning*, 46(1-3):225–254, 2002.

A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics*, 38(2):325–339, 1967.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Soceity, Series B*, 39(1):1–38, 1977.

J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

L. Didaci, G. Giacinto, F. Roli, and G. L. Marcialis. A study on the performances of dynamic classifier selection based on local accuracy estimation. *Pattern Recognition*, 38(11):2188–2191, 2005.

T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998.

T. G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the 1st International Workshop on Multiple Classifier Systems*, pages 1–15, Sardinia, Italy, 2000a.

T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2000b.

T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

T. G. Dietterich, G. Hao, and A. Ashenfelter. Gradient tree boosting for training conditional random fields. *Journal of Machine Learning Research*, 9: 2113–2139, 2008.

C. Domingo and O. Watanabe. Madaboost: A modification of AdaBoost. In *Proceedings of the 13th Annual Conference on Computational Learning Theory*, pages 180–189, Palo Alto, CA, 2000.

P. Domingos. Knowledge discovery via multiple models. *Intelligent Data Analysis*, 2(1-4):187–202, 1998.

P. Domingos. MetaCost: A general method for making classifiers cost-sensitive. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 155–164, San Diego, CA, 1999.

P. Domingos and M. Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–137, 1997.

C. Drummond and R. C. Holte. Exploiting the cost of (in)sensitivity of decision tree splitting criteria. In *Proceedings of the 17th International Conference on Machine Learning*, pages 239–246, San Francisco, CA, 2000.

C. Drummond and R. C. Holte. Cost curves: An improved method for visualizing classifier performance. *Machine Learning*, 65(1):95–130, 2006.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley, New York, NY, 2nd edition, 2000.

B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, New York, NY, 1993.

C. Elkan. The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 973–978, Seattle, WA, 2001.

S. Escalera, O. Pujol, and P. Radeva. Boosted landmarks of contextual descriptors and Forest-ECOC: A novel framework to detect and classify objects in clutter scenes. *Pattern Recognition Letters*, 28(13):1759–1768, 2007.

S. Escalera, O. Pujol, and P. Radeva. Error-correcting ouput codes library. *Journal of Machine Learning Research*, 11:661–664, 2010a.

S. Escalera, O. Pujol, and P. Radeva. On the decoding process in ternary error-correcting output codes. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 32(1):120–134, 2010b.

A. Estabrooks, T. Jo, and N. Japkowicz. A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence*, 20(1): 18–36, 2004.

M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231, Portland, OR, 1996.

V. Estivill-Castro. Why so many clustering algorithms - A position paper. *SIGKDD Explorations*, 4(1):65–75, 2002.

W. Fan. On the optimality of probability estimation by random decision trees. In *Proceedings of the 19th National Conference on Artificial Intelligence*, pages 336–341, San Jose, CA, 2004.

W. Fan, S. J. Stolfo, J. Zhang, and P. K. Chan. AdaCost: Misclassification cost-sensitive boosting. In *Proceedings of the 16th International Conference on Machine Learning*, pages 97–105, Bled, Slovenia, 1999.

W. Fan, F. Chu, H. Wang, and P. S. Yu. Pruning and dynamic scheduling of cost-sensitive ensembles. In *Proceedings of the 18th National Conference on Artificial Intelligence*, pages 146–151, Edmonton, Canada, 2002.

W. Fan, H. Wang, P. S. Yu, and S. Ma. Is random model better? On its accuracy and efficiency. In *Proceedings of the 3rd IEEE International Conferenceon Data Mining*, pages 51–58, Melbourne, FL, 2003.

R. Fano. *Transmission of Information: Statistical Theory of Communications*. MIT Press, Cambridge, MA, 1961.

T. Fawcett. ROC graphs with instance varying costs. *Pattern Recognition Letters*, 27(8):882–891, 2006.

X. Z. Fern and C. E. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proceedings of the 20th International Conference on Machine Learning*, pages 186–193, Washington, DC, 2003.

X. Z. Fern and C. E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004.

X. Z. Fern and W. Lin. Cluster ensemble selection. In *Proceedings of the 8th SIAM International Conference on Data Mining*, pages 787–797, Atlanta, GA, 2008.

C. Ferri, J. Hernández-Orallo, and M. J. Ramírez-Quintana. From ensemble methods to comprehensible models. In *Proceedings of the 5th International Conference on Discovery Science*, pages 165–177, Lübeck, Germany, 2002.

D. Fisher. Improving inference through conceptual clustering. In *Proceedings of the 6th National Conference on Artificial Intelligence*, pages 461–465, Seattle, WA, 1987.

J. L. Fleiss. *Statistical Methods for Rates and Proportions*. John Wiley & Sons, New York, NY, 2nd edition, 1981.

E. Frank and M. Hall. Visualizing class probability estimators. In *Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 168–179, Cavtat-Dubrovnik, Croatia, 2003.

A. Fred and A. K. Jain. Data clustering using evidence accumulation. In *Proceedings of the 16th International Conference on Pattern Recognition*, pages 276–280, Quebec, Canada, 2002.

A. Fred and A. K. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):835–850, 2005.

Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, 1995.

Y. Freund. An adaptive version of the boost by majority algorithm. *Machine Learning*, 43(3):293–318, 2001.

Y. Freund. A more robust boosting algorithm. CORR abs/0905.2138, 2009.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the 2nd European Conference on Computational Learning Theory*, pages 23–37, Barcelona, Spain, 1995.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, 1997.

J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting (with discussions). *Annals of Statistics*, 28(2):337–407, 2000.

J. H. Friedman and P. Hall. On bagging and nonlinear estimation. *Journal of Statistical Planning and Inference*, 137(3):669–683, 2007.

J. H. Friedman and W. Stuetzle. Projection pursuit regression. *Journal of American Statistical Association*, 76(376):817–823, 1981.

N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2):131–163, 1997.

G. Fumera and F. Roli. A theoretical and experimental analysis of linear combiners for multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):942–956, 2005.

W. Gao and Z.-H. Zhou. Approximation stability and boosting. In *Proceedings of the 21st International Conference on Algorithmic Learning Theory*, pages 59–73, Canberra, Australia, 2010a.

W. Gao and Z.-H. Zhou. On the doubt about margin explanation of boosting. CORR abs/1009.3613, 2012.

C. W. Gardiner. *Handbook of Stochastic Methods*. Springer, New York, NY, 3rd edition, 2004.

S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.

P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006.

G. Giacinto and F. Roli. Adaptive selection of image classifiers. In *Proceedings of the 9th International Conference on Image Analysis and Processing*, pages 38–45, Florence, Italy, 1997.

G. Giacinto and F. Roli. A theoretical framework for dynamic classifier selection. In *Proceedings of the 15th International Conference on Pattern Recognition*, pages 2008–2011, Barcelona, Spain, 2000a.

G. Giacinto and F. Roli. Dynamic classifier selection. In *Proceedings of the 1st International Workshop on Multiple Classifier Systems*, pages 177–189, Cagliari, Italy, 2000b.

G. Giacinto and F. Roli. Design of effective neural network ensembles for image classification purposes. *Image and Vision Computing*, 19(9-10): 699–707, 2001.

G. Giacinto, F. Roli, and G. Fumera. Design of effective multiple classifier systems by clustering of classifiers. In *Proceedings of the 15th International Conference on Pattern Recognition*, pages 160–163, Barcelona, Spain, 2000.

G. Giacinto, F. Roli, and L. Didaci. Fusion of multiple classifiers for intrusion detection in computer networks. *Pattern Recognition Letters*, 24(12): 1795–1803, 2003.

G. Giacinto, R. Perdisci, M. D. Rio, and F. Roli. Intrusion detection in computer networks by a modular ensemble of one-class classifiers. *Information Fusion*, 9(1):69–82, 2008.

T. Gneiting and A. E. Raftery. Atmospheric science: Weather forecasting with ensemble methods. *Science*, 310(5746):248–249, 2005.

K. Goebel, M. Krok, and H. Sutherland. Diagnostic information fusion: Requirements flowdown and interface issues. In *Proceedings of the IEEE Aerospace Conference*, volume 6, pages 155–162, Big Sky, MT, 2000.

D. E. Goldberg. *Genetic Algorithm in Search, Optimization and Machine Learning*. Addison-Wesley, Boston, MA, 1989.

D. M. Green and J. M. Swets. *Signal Detection Theory and Psychophysics*. John Wiley & Sons, New York, NY, 1966.

A. J. Grove and D. Schuurmans. Boosting in the limit: Maximizing the margin of learned ensembles. In *Proceedings of the 15th National Conference on Artificial Intelligence*, pages 692–699, Madison, WI, 1998.

S. Guha, R. Rastogi, and K. Shim. ROCK: A robust clustering algorithm for categorical attributes. In *Proceedings of the 15th International Conference on Data Engineering*, pages 512–521, Sydney, Australia, 1999.

H. Guo and H. L. Viktor. Learning from imbalanced data sets with boosting and data generation: The DataBoost-IM approach. *SIGKDD Explorations*, 6(1):30–39, 2004.

Y. Guo and D. Schuurmans. Discriminative batch mode active learning. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 593–600. MIT Press, Cambridge, MA, 2008.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

S. T. Hadjitodorov and L. I. Kuncheva. Selecting diversifying heuristics for cluster ensembles. In *Proceedings of the 7th International Workshop on Multiple Classifier Systems*, pages 200–209, Prague, Czech, 2007.

S. T. Hadjitodorov, L. I. Kuncheva, and L. P. Todorova. Moderate diversity for better cluster ensembles. *Information Fusion*, 7(3):264–275, 2006.

M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2-3):107–145, 2001.

J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, San Francisco, CA, 2nd edition, 2006.

D. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining*. MIT Press, Cambridge, MA, 2001.

D. J. Hand. Measuring classifier performance: A coherent alternative to the area under the ROC curve. *Machine Learning*, 77(1):103–123, 2009.

D. J. Hand and R. J. Till. A simple generalization of the area under the ROC curve to multiple classification problems. *Machine Learning*, 45(2):171–186, 2001.

J. A. Hanley and B. J. McNeil. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology*, 148(3):839–843, 1983.

L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990.

M. Harries. Boosting a strong learner: Evidence against the minimum margin. In *Proceedings of the 16th International Conference on Machine Learning*, pages 171–179, Bled, Slovenia, 1999.

T. Hastie and R. Tibshirani. Classification by pairwise coupling. *Annals of Statistics*, 26(2):451–471, 1998.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, New York, NY, 2001.

S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice-Hall, Upper Saddle River, NJ, 2nd edition, 1998.

H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

Z. He, X. Xu, and S. Deng. A cluster ensemble method for clustering categorical data. *Information Fusion*, 6(2):143–151, 2005.

M. Hellman and J. Raviv. Probability of error, equivocation, and the Chernoff bound. *IEEE Transactions on Information Theory*, 16(4):368–372, 1970.

D. Hernández-Lobato, G. Martínez-Muñoz, and A. Suárez. Statistical instance-based pruning in ensembles of independent classifiers. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 31(2):364–369, 2009.

D. Hernández-Lobato, G. Martínez-Muñoz, and A. Suárez. Empirical analysis and evaluation of approximate techniques for pruning regression bagging ensembles. *Neurocomputing*, 74(12-13):2250–2264, 2011.

A. Hinneburg and D. A. Keim. An efficient approach to clustering in large multimedia databases with noise. In *Proceedings of the 4th International*

Conference on Knowledge Discovery and Data Mining, pages 58–65, New York, NY, 1998.

T. K. Ho. Random decision forests. In *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, pages 278–282, Montreal, Canada, 1995.

T. K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8): 832–844, 1998.

T. K. Ho, J. J. Hull, and S. N. Srihari. Decision combination in multiple classifier systems. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 16(1):66–75, 1994.

V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004.

S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu. Semisupervised SVM batch mode active learning with applications to image retrieval. *ACM Transactions on Information Systems*, 27(3):1–29, 2009.

Y. Hong, S. Kwong, H. Wang, and Q. Ren. Resampling-based selective clustering ensembles. *Pattern Recognition Letters*, 41(9):2742–2756, 2009.

P. Hore, L. Hall, and D. Goldgof. A cluster ensemble framework for large data sets. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 3342–3347, Taipei, Taiwan, ROC, 2006.

P. Hore, L. O. Hall, and D. B. Goldgof. A scalable framework for cluster ensembles. *Pattern Recognition*, 42(5):676–688, 2009.

C.-W. Hsu and C.-J. Lin. A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.

X. Hu, E. K. Park, and X. Zhang. Microarray gene cluster identification and annotation through cluster ensemble and EM-based informative textual summarization. *IEEE Transactions on Information Technology in Biomedicine*, 13(5):832–840, 2009.

F.-J. Huang, Z.-H. Zhou, H.-J. Zhang, and T. Chen. Pose invariant face recognition. In *Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition*, pages 245–250, Grenoble, France, 2000.

S.-J. Huang, R. Jin, and Z.-H. Zhou. Active learning by querying informative and representative examples. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Informa-*

tion Processing Systems 23, pages 892–900. MIT Press, Cambridge, MA, 2010.

Y. S. Huang and C. Y. Suen. A method of combining multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):90–94, 1995.

Z. Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2(3): 283–304, 1998.

R. A. Hutchinson, L.-P. Liu, and T. G. Dietterich. Incorporating boosted regression trees into ecological latent variable models. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, pages 1343–1348, San Francisco, CA, 2011.

R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, Upper Saddle River, NJ, 1988.

A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

G. M. James. Variance and bias for general loss functions. *Machine Learning*, 51(2):115–135, 2003.

T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the 16th International Conference on Machine Learning*, pages 200–209, Bled, Slovenia, 1999.

T. Joachims. A support vector method for multivariate performance measures. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 384–391, Bonn, Germany, 2005.

I. T. Jolliffe. *Principal Component Analysis*. Springer, New York, NY, 2nd edition, 2002.

M. I. Jordan and R. A. Jacobs. Hierarchies of adaptive experts. In J. E. Moody, S. J. Hanson, and R. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, pages 985–992. Morgan Kaufmann, San Francisco, CA, 1992.

M. I. Jordan and L. Xu. Convergence results for the EM approach to mixtures of experts architectures. *Neural Networks*, 8(9):1409–1431, 1995.

G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1): 359–392, 1998.

G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. Multilevel hypergraph partitioning: Application in VLSI domain. In *Proceedings of the 34th Annual Design Automation Conference*, pages 526–529, Anaheim, CA, 1997.

L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, New York, NY, 1990.

M. Kearns. Efficient noise tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998.

M. Kearns and L. G. Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pages 433–444, Seattle, WA, 1989.

M. J. Kearns and U. V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, MA, 1994.

J. Kittler and F. M. Alkoot. Sum versus vote fusion in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25 (1):110–115, 2003.

J. Kittler, M. Hatef, R. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

E. M. Kleinberg. On the algorithmic implementation of stochastic discrimination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(5):473–490, 2000.

D. E. Knuth. *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley, Reading, MA, 2nd edition, 1997.

A. H. Ko, R. Sabourin, and J. A. S. Britto. From dynamic classifier selection to dynamic ensemble selection. *Pattern Recognition*, 41(5):1718–1731, 2008.

R. Kohavi and D. H. Wolpert. Bias plus variance decomposition for zero-one loss functions. In *Proceedings of the 13th International Conference on Machine Learning*, pages 275–283, Bari, Italy, 1996.

T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, Berlin, 3rd edition, 1989.

J. F. Kolen and J. B. Pollack. Back propagation is sensitive to initial conditions. In R. Lippmann, J. E. Moody, and D. S. Touretzky, editors, *Advances in Neural Information Processing Systems 3*, pages 860–867. Morgan Kaufmann, San Francisco, CA, 1991.

J. Z. Kolter and M. A. Maloof. Learning to detect and classify malicious executables in the wild. *Journal of Machine Learning Research*, 7:2721–2744, 2006.

E. B. Kong and T. G. Dietterich. Error-correcting output coding corrects bias and variance. In *Proceedings of the 12th International Conference on Machine Learning*, pages 313–321, Tahoe City, CA, 1995.

A. Krogh and J. Vedelsby. Neural network ensembles, cross validation, and active learning. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 231–238. MIT Press, Cambridge, MA, 1995.

M. Kubat and S. Matwin. Addressing the curse of imbalanced training sets: One sided selection. In *Proceedings of the 14th Intenational Conference on Machine Learning*, pages 179–186, Nashville, TN, 1997.

H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–79, 1955.

M. Kukar and I. Kononenko. Cost-sensitive learning with neural networks. In *Proceedings of the 13th European Conference on Artificial Intelligence*, pages 445–449, Brighton, UK, 1998.

L. I. Kuncheva. A theoretical study on six classifier fusion strategies. *IEEE Transations on Pattern Analysis and Machine Intelligence*, 24(2):281–286, 2002.

L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley & Sons, Hoboken, NJ, 2004.

L. I. Kuncheva. Classifier ensembles: Facts, fiction, faults and future, 2008. Plenary Talk at the 19th International Conference on Pattern Recognition.

L. I. Kuncheva and S. T. Hadjitodorov. Using diversity in cluster ensembles. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 1214–1219, Hague, The Netherlands, 2004.

L. I. Kuncheva and D. P. Vetrov. Evaluation of stability of k-means cluster ensembles with respect to random initialization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1798–1808, 2006.

L. I. Kuncheva and C. J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207, 2003.

L. I. Kuncheva, J. C. Bezdek, and R. P. Duin. Decision templates for multiple classifier fusion: An experimental comparison. *Pattern Recognition*, 34 (2):299–314, 2001.

L. I. Kuncheva, C. J. Whitaker, C. Shipp, and R. Duin. Limits on the majority vote accuracy in classifier fusion. *Pattern Analysis and Applications*, 6(1): 22–31, 2003.

L. I. Kuncheva, S. T. Hadjitodorov, and L. P. Todorova. Experimental comparison of cluster ensemble methods. In *Proceedings of the 9th International Conference on Information Fusion*, pages 1–7, Florence, Italy, 2006.

S. Kutin and P. Niyogi. Almost-everywhere algorithmic stability and generalization error. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pages 275–282, Edmonton, Canada, 2002.

S. W. Kwok and C. Carter. Multiple decision trees. In *Proceedings of the 4th International Conference on Uncertainty in Artificial Intelligence*, pages 327–338, New York, NY, 1988.

L. Lam and S. Y. Suen. Application of majority voting to pattern recognition: An analysis of its behavior and performance. *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, 27(5):553–568, 1997.

A. Lazarevic and Z. Obradovic. Effective pruning of neural network classifier ensembles. In *Proceedings of the IEEE/INNS International Joint Conference on Neural Networks*, pages 796–801, Washington, DC, 2001.

D. Lewis and W. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12, Dublin, Ireland, 1994.

M. Li and Z.-H. Zhou. Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, 37(6):1088–1098, 2007.

M. Li, W. Wang, and Z.-H. Zhou. Exploiting remote learners in internet environment with agents. *Science China: Information Sciences*, 53(1):64–76, 2010.

N. Li and Z.-H. Zhou. Selective ensemble under regularization framework. In *Proceedings of the 8th International Workshop Multiple Classifier Systems*, pages 293–303, Reykjavik, Iceland, 2009.

S. Z. Li, Q. Fu, L. Gu, B. Schölkopf, and H. J. Zhang. Kernel machine based learning for multi-view face detection and pose estimation. In *Proceedings of the 8th International Conference on Computer Vision*, pages 674–679, Vancouver, Canada, 2001.

R. Liere and P. Tadepalli. Active learning with committees for text categorization. In *Proceedings of the 14th National Conference on Artificial Intelligence*, pages 591–596, Providence, RI, 1997.

H.-T. Lin and L. Li. Support vector machinery for infinite ensemble learning. *Journal of Machine Learning Research*, 9:285–312, 2008.

X. Lin, S. Yacoub, J. Burns, and S. Simske. Performance analysis of pattern classifier combination by plurality voting. *Pattern Recognition Letters*, 24 (12):1959–1969, 2003.

Y. M. Lin, Y. Lee, and G. Wahba. Support vector machines for classification in nonstandard situations. *Machine Learning*, 46(1):191–202, 2002.

F. T. Liu, K. M. Ting, and W. Fan. Maximizing tree diversity by building complete-random decision trees. In *Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 605–610, Hanoi, Vietnam, 2005.

F. T. Liu, K. M. Ting, Y. Yu, and Z.-H. Zhou. Spectrum of variable-random trees. *Journal of Artificial Intelligence Research*, 32(1):355–384, 2008a.

F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *Proceedings of the 8th IEEE International Conference on Data Mining*, pages 413–422, Pisa, Italy, 2008b.

F. T. Liu, K. M. Ting, and Z.-H. Zhou. On detecting clustered anomalies using SCiForest. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 274–290, Barcelona, Spain, 2010.

X.-Y. Liu and Z.-H. Zhou. The influence of class imbalance on cost-sensitive learning: An empirical study. In *Proceedings of the 6th IEEE International Conference on Data Mining*, pages 970–974, Hong Kong, China, 2006.

X.-Y. Liu and Z.-H. Zhou. Learning with cost intervals. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 403–412, Washington, DC, 2010.

X.-Y. Liu, J. Wu, and Z.-H. Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 39(2):539–550, 2009.

Y. Liu and X. Yao. Ensemble learning via negative correlation. *Neural Networks*, 12(10):1399–1404, 1999.

S. P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):128–137, 1982.

J. M. Lobo, A. Jiménez-Valverde, and R. Real. AUC: A misleading measure of the performance of predictive distribution models. *Global Ecology and Biogeography*, 17(2):145–151, 2008.

B. Long, Z. Zhang, and P. S. Yu. Combining multiple clusterings by soft correspondence. In *Proceedings of the 4th IEEE International Conference on Data Mining*, pages 282–289, Brighton, UK, 2005.

P. K. Mallapragada, R. Jin, A. K. Jain, and Y. Liu. Semiboost: Boosting for semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):2000–2014, 2009.

I. Maqsood, M. R. Khan, and A. Abraham. An ensemble of neural networks for weather forecasting. *Neural Computing & Applications*, 13(2):112–122, 2004.

D. D. Margineantu and T. G. Dietterich. Pruning adaptive boosting. In *Proceedings of the 14th International Conference on Machine Learning*, pages 211–218, Nashville, TN, 1997.

H. Markowitz. Portfolio selection. *Journal of Finance*, 7(1):77–91, 1952.

G. Martínez-Muñoz and A. Suárez. Aggregation ordering in bagging. In *Proceedings of the IASTED International Conference on Artifical Intelligence and Applications*, pages 258–263, Innsbruck, Austria, 2004.

G. Martínez-Muñoz and A. Suárez. Pruning in ordered bagging ensembles. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 609–616, Pittsburgh, PA, 2006.

G. Martínez-Muñoz and A. Suárez. Using boosting to prune bagging ensembles. *Pattern Recognition Letters*, 28(1):156–165, 2007.

G. Martínez-Muñoz, D. Hernández-Lobato, and A. Suárez. An analysis of ensemble pruning techniques based on ordered aggregation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 31(2):245–259, 2009.

H. Masnadi-Shirazi and N. Vasconcelos. Asymmetric Boosting. In *Proceedings of the 24th International Conference on Machine Learning*, pages 609–616, Corvallis, OR, 2007.

L. Mason, J. Baxter, P. L. Bartlett, and M. Frean. Functional gradient techniques for combining hypotheses. In P. J. Bartlett, B. Schölkopf, D. Schuurmans, and A. J. Smola, editors, *Advances in Large-Margin Classifiers*, pages 221–246. MIT Press, Cambridge, MA, 2000.

A. Maurer and M. Pontil. Empirical Bernstein bounds and sample-variance penalization. In *Proceedings of the 22nd Conference on Learning Theory*, Montreal, Canada, 2009.

A. McCallum and K. Nigam. Employing EM and pool-based active learning for text classification. In *Proceedings of the 15th International Conference on Machine Learning*, pages 350–358, Madison, WI, 1998.

R. A. McDonald, D. J. Hand, and I. A. Eckley. An empirical comparison of three boosting algorithms on real data sets with artificial class noise. In *Proceedings of the 4th International Workshop on Multiple Classifier Systems*, pages 35–44, Guilford, UK, 2003.

W. McGill. Multivariate information transmission. *IEEE Transactions on Information Theory*, 4(4):93–111, 1954.

D. Mease and A. Wyner. Evidence contrary to the statistical view of boosting (with discussions). *Journal of Machine Learning Research*, 9:131–201, 2008.

N. Meinshausen and P. Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B*, 72(4):417–473, 2010.

P. Melville and R. J. Mooney. Creating diversity in ensembles using artificial data. *Information Fusion*, 6(1):99–111, 2005.

C. E. Metz. Basic principles of ROC analysis. *Seminars in Nuclear Medicine*, 8(4):283–298, 1978.

D. J. Miller and H. S. Uyar. A mixture of experts classifier with learning based on both labelled and unlabelled data. In M. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 571–577. MIT Press, Cambridge, MA, 1997.

T. M. Mitchell. *Machine Learning*. McGraw-Hill, New York, NY, 1997.

X. Mu, P. Watta, and M. H. Hassoun. Analysis of a plurality voting-based combination of classifiers. *Neural Processing Letters*, 29(2):89–107, 2009.

I. Mukherjee and R. Schapire. A theory of multiclass boosting. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1714–1722. MIT Press, Cambridge, MA, 2010.

S. K. Murthy, S. Kasif, and S. Salzberg. A system for the induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–33, 1994.

A.M. Narasimhamurthy. A framework for the analysis of majority voting. In *Proceedings of the 13th Scandinavian Conference on Image Analysis*, pages 268–274, Halmstad, Sweden, 2003.

A. Narasimhamurthy. Theoretical bounds of majority voting performance for a binary classification problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1988–1995, 2005.

S. Nash and A. Sofer. *Linear and Nonlinear Programming*. McGraw-Hill, New York, NY, 1996.

R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 144–155, Santiago, Chile, 1994.

H. T. Nguyen and A. W. M. Smeulders. Active learning using pre-clustering. In *Proceedings of the 21st International Conference on Machine Learning*, pages 623–630, Banff, Canada, 2004.

K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2-3): 103–134, 2000.

N. J. Nilsson. *Learning Machines: Foundations of Trainable Pattern-Classifying Systems*. McGraw-Hill, New York, NY, 1965.

D. Opitz and R. Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198, 1999.

S. Panigrahi, A. Kundu, S. Sural, and A. K. Majumdar. Credit card fraud detection: A fusion approach using Dempster-Shafer theory and Bayesian learning. *Information Fusion*, 10(4):354–363, 2009.

I. Partalas, G. Tsoumakas, and I. Vlahavas. Pruning an ensemble of classifiers via reinforcement learning. *Neurocomputing*, 72(7-9):1900–1909, 2009.

D. Partridge and W. J. Krzanowski. Software diversity: Practical statistics for its measurement and exploitation. *Information & Software Technology*, 39(10):707–717, 1997.

A. Passerini, M. Pontil, and P. Frasconi. New results on error correcting output codes of kernel machines. *IEEE Transactions on Neural Networks*, 15 (1):45–54, 2004.

M. P. Perrone and L. N. Cooper. When networks disagree: Ensemble method for neural networks. In R. J. Mammone, editor, *Artificial Neural Networks for Spech and Vision*, pages 126–142. Chapman & Hall, New York, NY, 1993.

J. C. Platt. Probabilities for SV machines. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, Cambridge, MA, 2000.

R. Polikar, A. Topalis, D. Parikh, D. Green, J. Frymiare, J. Kounios, and C. M. Clark. An ensemble based data fusion approach for early diagnosis of Alzheimer's disease. *Information Fusion*, 9(1):83–95, 2008.

B. R. Preiss. *Data Structures and Algorithms with Object-Oriented Design Patterns in Java*. Wiley, Hoboken, NJ, 1999.

O. Pujol, P. Radeva, and J. Vitrià. Discriminant ECOC: A heuristic method for application dependent design of error correcting output codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):1007–1012, 2006.

O. Pujol, S. Escalera, and P. Radeva. An incremental node embedding technique for error correcting output codes. *Pattern Recognition*, 41(2):713–725, 2008.

J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco, CA, 1993.

J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1998.

Š. Raudys and F. Roli. The behavior knowledge space fusion method: Analysis of generalization error and strategies for performance improvement. In *Proceedings of the 4th International Workshop on Multiple Classifier Systems*, pages 55–64, Guildford, UK, 2003.

R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26(2):195–239, 1984.

L. Reyzin and R. E. Schapire. How boosting the margin can also boost classifier complexity. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 753–760, Pittsburgh, PA, 2006.

B. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, UK, 1996.

M. Robnik-Šikonja. Improving random forests. In *Proceedings of the 15th European Conference on Machine Learning*, pages 359–370, Pisa, Italy, 2004.

J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso. Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1619–1630, 2006.

G. Rogova. Combining the results of several neural network classifiers. *Neural Networks*, 7(5):777–781, 1994.

L. Rokach. *Pattern Classification Using Ensemble Methods*. World Scientific, Singapore, 2010.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, pages 318–362. MIT Press, Cambridge, MA, 1986.

D. Ruta and B. Gabrys. Application of the evolutionary algorithms for classifier selection in multiple classifier systems with majority voting. In *Proceedings of the 2nd International Workshop on Multiple Classifier Systems*, pages 399–408, Cambridge, UK, 2001.

R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2): 197–227, 1990.

R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of Statistics*, 26(5):1651–1686, 1998.

J. Schiffers. A classification approach incorporating misclassification costs. *Intelligent Data Analysis*, 1(1):59–68, 1997.

B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors. *Advances in Kernel Methods: Support Vector Learning*. MIT Press, Cambridge, MA, 1999.

M. G. Schultz, E. Eskin, E. Zadok, and S. J. Stolfo. Data mining methods for detection of new malicious executables. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 38–49, Oakland, CA, 2001.

A. K. Seewald. How to make stacking better and faster while also taking care of an unknown weakness. In *Proceedings of the 19th International Conference on Machine Learning*, pages 554–561, Sydney, Australia, 2002.

B. Settles. Active learning literature survey. Technical Report 1648, Department of Computer Sciences, University of Wisconsin at Madison, Madison, WI, 2009.

H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the 5th Annual ACM Conference on Computational Learning Theory*, pages 287–294, Pittsburgh, PA, 1992.

G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, NJ, 1976.

G. Sheikholeslami, S. Chatterjee, and A. Zhang. WaveCluster: A multi-resolution clustering approach for very large spatial databases. In *Proceedings of the 24th International Conference on Very Large Data Bases*, pages 428–439, New York, NY, 1998.

H. B. Shen and K. C. Chou. Ensemble classifier for protein fold pattern recognition. *Bioinformatics*, 22(14):1717–1722, 2006.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

C. A. Shipp and L. I. Kuncheva. Relationships between combination methods and measures of diversity in combining classifiers. *Information Fusion*, 3(2):135–148, 2002.

D. B. Skalak. The sources of increased accuracy for two proposed boosting algorithms. In *Working Notes of the AAAI'96 Workshop on Integrating Multiple Learned Models*, Portland, OR, 1996.

N. Slonim, N. Friedman, and N. Tishby. Multivariate information bottleneck. *Neural Computation*, 18(8):1739–1789, 2006.

P. Smyth and D. Wolpert. Stacked density estimation. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 668–674. MIT Press, Cambridge, MA, 1998.

P. H. A. Sneath and R. R. Sokal. *Numerical Taxonomy: The Principles and Practice of Numerical Classification*. W. H. Freeman, San Francisco, CA, 1973.

V. Soto, G. Martínez-Muñoz, D. Hernández-Lobato, and A. Suárez. A double pruning algorithm for classification ensembles. In *Proceedings of 9th International Workshop Multiple Classifier Systems*, pages 104–113, Cairo, Egypt, 2010.

K. A. Spackman. Signal detection theory: Valuable tools for evaluating inductive learning. In *Proceedings of the 6th International Workshop on Machine Learning*, pages 160–163, Ithaca, NY, 1989.

A. Strehl and J. Ghosh. Cluster ensembles - A knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

A. Strehl, J. Ghosh, and R. J. Mooney. Impact of similarity measures on web-page clustering. In *Proceedings of the AAAI'2000 Workshop on AI for Web Search*, pages 58–64, Austin, TX, 2000.

M. Studeny and J. Vejnarova. The multi-information function as a tool for measuring stochastic dependence. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 261–298. Kluwer, Norwell, MA, 1998.

Y. Sun, A. K. C. Wong, and Y. Wang. Parameter inference of cost-sensitive boosting algorithms. In *Proceedings of the 4th International Conference on Machine Learning and Data Mining in Pattern Recognition*, pages 21–30, Leipzig, Germany, 2005.

C. Tamon and J. Xiang. On the boosting pruning problem. In *Proceedings of the 11th European Conference on Machine Learning*, pages 404–412, Barcelona, Spain, 2000.

A. C. Tan, D. Gilbert, and Y. Deville. Multi-class protein fold classification using a new ensemble machine learning approach. *Genome Informatics*, 14:206–217, 2003.

P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, Upper Saddle River, NJ, 2006.

E. K. Tang, P. N. Suganthan, and X. Yao. An analysis of diversity measures. *Machine Learning*, 65(1):247–271, 2006.

J. W. Taylor and R. Buizza. Neural network load forecasting with weather ensemble predictions. *IEEE Transactions on Power Systems*, 17(3):626–632, 2002.

S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, New York, NY, 4th edition, 2009.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288, 1996a.

R. Tibshirani. Bias, variance and prediction error for classification rules. Technical report, Department of Statistics, University of Toronto, 1996b.

A. B. Tickle, R. Andrews, M. Golea, and J. Diederich. The truth will come to light: Directions and challenges in extracting the knowledge embedded within trained artificial neural networks. *IEEE Transactions on Neural Networks*, 9(6):1057–1067, 1998.

K. M. Ting. A comparative study of cost-sensitive boosting algorithms. In *Proceedings of the 17th International Conference on Machine Learning*, pages 983–990, San Francisco, CA, 2000.

K. M. Ting. An instance-weighting method to induce cost-sensitive trees. *IEEE Transactions on Knowledge and Data Engineering*, 14(3):659–665, 2002.

K. M. Ting and I. H. Witten. Issues in stacked generalization. *Journal of Artificial Intelligence Research*, 10:271–289, 1999.

I. Tomek. Two modifications of CNN. *IEEE Transactions on Systems, Man and Cybernetics*, 6(11):769–772, 1976.

S. Tong and D. Koller. Support vector machine active learning with applications to text classification. In *Proceedings of the 17th International Conference on Machine Learning*, pages 999–1006, San Francisco, CA, 2000.

A. Topchy, A. K. Jain, and W. Punch. Combining multiple weak clusterings. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, pages 331–338, Melbourne, FL, 2003.

A. Topchy, A. K. Jain, and W. Punch. A mixture model for clustering ensembles. In *Proceedings of the 4th SIAM International Conference on Data Mining*, pages 379–390, Lake Buena Vista, FL, 2004a.

A. Topchy, B. Minaei-Bidgoli, A. K. Jain, and W. F. Punch. Adaptive clustering ensembles. In *Proceedings of the 17th International Conference on Pattern Recognition*, pages 272–275, Cambridge, UK, 2004b.

A. P. Topchy, M. H. C. Law, A. K. Jain, and A. L. Fred. Analysis of consensus partition in cluster ensemble. In *Proceedings of the 4th IEEE International Conference on Data Mining*, pages 225–232, Brighton, UK, 2004c.

G. Tsoumakas, I. Katakis, and I. Vlahavas. Effective voting of heterogeneous classifiers. In *Proceedings of the 15th European Conference on Machine Learning*, pages 465–476, Pisa, Italy, 2004.

G. Tsoumakas, L. Angelis, and I. P. Vlahavas. Selective fusion of heterogeneous classifiers. *Intelligent Data Analysis*, 9(6):511–525, 2005.

G. Tsoumakas, I. Partalas, and I. Vlahavas. An ensemble pruning primer. In O. Okun and G. Valentini, editors, *Applications of Supervised and Unsupervised Ensemble Methods*, pages 155–165. Springer, Berlin, 2009.

K. Tumer. *Linear and Order Statistics Combiners for Reliable Pattern Classification*. PhD thesis, The University of Texas at Austin, 1996.

K. Tumer and J. Ghosh. Theoretical foundations of linear and order statistics combiners for neural pattern classifiers. Technical Report TR-95-02-98, Computer and Vision Research Center, University of Texas, Austin, 1995.

K. Tumer and J. Ghosh. Analysis of decision boundaries in linearly combined neural classifiers. *Pattern Recognition*, 29(2):341–348, 1996.

P. D. Turney. Types of cost in inductive concept learning. In *Proceedings of the ICML'2000 Workshop on Cost-Sensitive Learning*, pages 15–21, San Francisco, CA, 2000.

N. Ueda and R. Nakano. Generalization error of ensemble estimators. In *Proceedings of the IEEE International Conference on Neural Networks*, pages 90–95, Washington, DC, 1996.

W. Utschick and W. Weichselberger. Stochastic organization of output codes in multiclass learning problems. *Neural Computation*, 13(5):1065–1102, 2004.

L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27 (11):1134–1142, 1984.

H. Valizadegan, R. Jin, and A. K. Jain. Semi-supervised boosting for multiclass classification. In *Proceedings of the 19th European Conference on Machine Learning*, pages 522–537, Antwerp, Belgium, 2008.

C. van Rijsbergen. *Information Retrieval*. Butterworths, London, 1979.

V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, NY, 1998.

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 511–518, Kauai, HI, 2001.

P. Viola and M. Jones. Fast and robust classification using asymmetric Adaboost and a detector cascade. In T. G. Dietterich, S. Becker, and

Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 1311–1318. MIT Press, Cambridge, MA, 2002.

P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

L. Wang, M. Sugiyama, C. Yang, Z.-H. Zhou, and J. Feng. On the margin explanation of boosting algorithm. In *Proceedings of the 21st Annual Conference on Learning Theory*, pages 479–490, Helsinki, Finland, 2008.

W. Wang and Z.-H. Zhou. On multi-view active learning and the combination with semi-supervised learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1152–1159, Helsinki, Finland, 2008.

W. Wang and Z.-H. Zhou. Multi-view active learning in the non-realizable case. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 2388–2396. MIT Press, Cambridge, MA, 2010.

W. Wang, J. Yang, and R. Muntz. STING: A statistical information grid approach to spatial data mining. In *Proceedings of the 23rd International Conference on Very Large Data Bases*, pages 186–195, Athens, Greece, 1997.

M. K. Warmuth, K. A. Glocer, and S. V. Vishwanathan. Entropy regularized LPBoost. In *Proceedings of the 19th International Conference on Algorithmic Learning Theory*, pages 256–271, Budapest, Hungary, 2008.

S. Watanabe. Information theoretical analysis of multivariate correlation. *IBM Journal of Research and Development*, 4(1):66–82, 1960.

S. Waterhouse, D. Mackay, and T. Robinson. Bayesian methods for mixtures of experts. In D. S. Touretzky, M. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pages 351–357. MIT Press, Cambridge, MA, 1996.

S. R. Waterhouse and A. J. Robinson. Constructive algorithms for hierarchical mixtures of experts. In D. S. Touretzky, M. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pages 584–590. MIT Press, Cambridge, MA, 1996.

C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3):279–292, 1992.

G. I. Webb and Z. Zheng. Multistrategy ensemble learning: Reducing error by combining ensemble learning techniques. *IEEE Transactions on Knowledge and Data Engineering*, 16(8):980–991, 2004.

G. I. Webb, J. R. Boughton, and Z. Wang. Not so naïve Bayes: Aggregating one-dependence estimators. *Machine Learning*, 58(1):5–24, 2005.

P. Werbos. *Beyond regression: New tools for prediction and analysis in the behavior science*. PhD thesis, Harvard University, Cambridge, MA, 1974.

D. West, S. Dellana, and J. Qian. Neural network ensemble strategies for financial decision applications. *Computers & Operations Research*, 32(10): 2543–2559, 2005.

T. Windeatt and R. Ghaderi. Coding and decoding strategies for multi-class learning problems. *Information Fusion*, 4(1):11–21, 2003.

D. H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–260, 1992.

D. H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341–1390, 1996.

D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

D. H. Wolpert and W. G. Macready. An efficient method to estimate bagging's generalization error. *Machine Learning*, 35(1):41–55, 1999.

K. Woods, W. P. Kegelmeyer, and K. Bowyer. Combination of multiple classifiers using local accuracy estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):405–410, 1997.

J. Wu, S. C. Brubaker, M. D. Mullin, and J. M. Rehg. Fast asymmetric learning for cascade face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):369–382, 2008.

L. Xu and S. Amari. Combining classifiers and learning mixture-of-experts. In J. R. R. Dopico, J. Dorado, and A. Pazos, editors, *Encyclopedia of Artificial Intelligence*, pages 318–326. IGI, Berlin, 2009.

L. Xu and M. I. Jordan. On convergence properties of the EM algorithm for Gaussian mixtures. *Neural Computation*, 8(1):129–151, 1996.

L. Xu, A. Krzyzak, and C. Y. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems Man and Cybernetics*, 22(3):418–435, 1992.

L. Xu, M. I. Jordan, and G. E. Hinton. An alternative model for mixtures of experts. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 633–640. MIT Press, Cambridge, MA, 1995.

L. Yan, R. H. Dodier, M. Mozer, and R. H. Wolniewicz. Optimizing classifier performance via an approximation to the Wilcoxon-Mann-Whitney statistic. In *Proceedings of the 20th International Conference on Machine Learning*, pages 848–855, Washington, DC, 2003.

W. Yan and F. Xue. Jet engine gas path fault diagnosis using dynamic fusion of multiple classifiers. In *Proceedings of the International Joint Conference on Neural Networks*, pages 1585–1591, Hong Kong, China, 2008.

Y. Yu, Y.-F. Li, and Z.-H. Zhou. Diversity regularized machine. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 1603–1608, Barcelona, Spain, 2011.

Z. Yu and H.-S. Wong. Class discovery from gene expression data based on perturbation and cluster ensemble. *IEEE Transactions on NanoBioscience*, 18(2):147–160, 2009.

G. Yule. On the association of attributes in statistics. *Philosophical Transactions of the Royal Society of London*, 194:257–319, 1900.

B. Zadrozny and C. Elkan. Learning and making decisions when costs and probabilities are both unknown. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 204–213, San Francisco, CA, 2001a.

B. Zadrozny and C. Elkan. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *Proceedings of the 18th International Conference on Machine Learning*, pages 609–616, Williamstown, MA, 2001b.

B. Zadrozny, J. Langford, and N. Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, pages 435–442, Melbourne, FL, 2003.

M.-L. Zhang and Z.-H. Zhou. Exploiting unlabeled data to enhance ensemble diversity. In *Proceedings of the 9th IEEE International Conference on Data Mining*, pages 609–618, Sydney, Australia, 2010.

T. Zhang. Analysis of regularized linear functions for classification problems. Technical Report RC-21572, IBM, 1999.

T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: An efficient data clustering method for very large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 103–114, Montreal, Canada, 1996.

X. Zhang, S. Wang, T. Shan, and L. Jiao. Selective SVMs ensemble driven by immune clonal algorithm. In *Proceedings of the EvoWorkshops*, pages 325–333, Lausanne, Switzerland, 2005.

X. Zhang, L. Jiao, F. Liu, L. Bo, and M. Gong. Spectral clustering ensemble applied to SAR image segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 46(7):2126–2136, 2008.

Y. Zhang, S. Burer, and W. N. Street. Ensemble pruning via semi-definite programming. *Journal of Machine Learning Research*, 7:1315–1338, 2006.

Z. Zheng and G. I. Webb. Laze learning of Bayesian rules. *Machine Learning*, 41(1):53–84, 2000.

D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

Z.-H. Zhou. Rule extraction: Using neural networks or for neural networks? *Journal of Computer Science and Technology*, 19(2):249–253, 2004.

Z.-H. Zhou. Comprehensibility of data mining algorithms. In J. Wang, editor, *Encyclopedia of Data Warehousing and Mining*, pages 190–195. IGI, Hershey, PA, 2005.

Z.-H. Zhou. When semi-supervised learning meets ensemble learning. In *Proceedings of the 8th International Workshop on Multiple Classifier Systems*, pages 529–538, Reykjavik, Iceland, 2009.

Z.-H. Zhou. When semi-supervised learning meets ensemble learning. *Frontiers of Electrical and Electronic Engineering in China*, 6(1):6–16, 2011.

Z.-H. Zhou and Y. Jiang. Medical diagnosis with C4.5 rule preceded by artificial neural network ensemble. *IEEE Transactions on Information Technology in Biomedicine*, 7(1):37–42, 2003.

Z.-H. Zhou and Y. Jiang. NeC4.5: Neural ensemble based C4.5. *IEEE Transactions on Knowledge and Data Engineering*, 16(6):770–773, 2004.

Z.-H. Zhou and M. Li. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17 (11):1529–1541, 2005.

Z.-H. Zhou and M. Li. Semi-supervised regression with co-training style algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 19 (11):1479–1493, 2007.

Z.-H. Zhou and M. Li. Semi-supervised learning by disagreement. *Knowledge and Information Systems*, 24(3):415–439, 2010a.

Z.-H. Zhou and N. Li. Multi-information ensemble diversity. In *Proceedings of the 9th International Workshop on Multiple Classifier Systems*, pages 134–144, Cairo, Egypt, 2010b.

Z.-H. Zhou and X.-Y. Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):63–77, 2006.

Z.-H. Zhou and X.-Y. Liu. On multi-class cost-sensitive learning. *Computational Intelligence*, 26(3):232–257, 2010.

Z.-H. Zhou and W. Tang. Selective ensemble of decision trees. In *Proceedings of the 9th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*, pages 476–483, Chongqing, China, 2003.

Z.-H. Zhou and W. Tang. Clusterer ensemble. *Knowledge-Based Systems*, 19 (1):77–83, 2006.

Z.-H. Zhou and Y. Yu. Ensembling local learners through multimodal perturbation. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 35(4):725–735, 2005.

Z.-H. Zhou, Y. Jiang, Y.-B. Yang, and S.-F. Chen. Lung cancer cell identification based on artificial neural network ensembles. *Artificial Intelligence in Medicine*, 24(1):25–36, 2002a.

Z.-H. Zhou, J. Wu, and W. Tang. Ensembling neural networks: Many could be better than all. *Artificial Intelligence*, 137(1-2):239–263, 2002b.

Z.-H. Zhou, Y. Jiang, and S.-F. Chen. Extracting symbolic rules from trained neural network ensembles. *AI Communications*, 16(1):3–15, 2003.

Z.-H. Zhou, K.-J. Chen, and H.-B. Dai. Enhancing relevance feedback in image retrieval using unlabeled data. *ACM Transactions on Information Systems*, 24(2):219–244, 2006.

J. Zhu, S. Rosset, H. Zou, and T. Hastie. Multi-class AdaBoost. Technical report, Department of Statistics, University of Michigan, Ann Arbor, MI, 2006.

X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Department of Computer Sciences, University of Wisconsin at Madison, Madison, WI, 2006. http://www.cs.wisc.edu/∼jerryzhu/pub/ssl_survey.pdf.

X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning*, pages 912–919, Washington, DC, 2003.

X. Zhu, X. Wu, and Y. Yang. Dynamic classifier selection for effective mining from noisy data streams. In *Proceedings of the 14th IEEE International Conference on Data Mining*, pages 305–312, Brighton, UK, 2004.

Chapman & Hall/CRC
Machine Learning & Pattern Recognition Series

"Professor Zhou's book is a comprehensive introduction to ensemble methods in machine learning. It reviews the latest research in this exciting area. I learned a lot reading it!"
—Thomas G. Dietterich, Oregon State University, ACM Fellow, and founding president of the International Machine Learning Society

"This is a timely book. Right time and right book … with an authoritative but inclusive style that will allow many readers to gain knowledge on the topic."
—Fabio Roli, University of Cagliari

An up-to-date, self-contained introduction to a state-of-the-art machine learning approach, **Ensemble Methods: Foundations and Algorithms** shows how these accurate methods are used in real-world tasks. It gives you the necessary groundwork to carry out further research in this evolving field.

Features

- Supplies the basics for readers unfamiliar with machine learning and pattern recognition
- Covers nearly all aspects of ensemble techniques such as combination methods and diversity generation methods
- Presents the theoretical foundations and extensions of many ensemble methods, including Boosting, Bagging, Random Trees, and Stacking
- Introduces the use of ensemble methods in computer vision, computer security, medical imaging, and famous data mining competitions
- Highlights future research directions
- Provides additional reading sections in each chapter and references at the back of the book

K11467

ISBN: 978-1-4398-3003-1
90000
9 781439 830031

CRC Press
Taylor & Francis Group
an informa business
www.crcpress.com

6000 Broken Sound Parkway, NW
Suite 300, Boca Raton, FL 33487
711 Third Avenue
New York, NY 10017
2 Park Square, Milton Park
Abingdon, Oxon OX14 4RN, UK