Available online at www.sciencedirect.com
SCIENCE DIRECT
Neural Networks 19 (2006) 113–121
Neural Networks
www.elsevier.com/locate/neunet

2006 Special issue

Computational intelligence in earth sciences and environmental applications: Issues and challenges

V. Cherkassky [a,*], V. Krasnopolsky [b,c], D.P. Solomatine [d], J. Valdes [e]

[a] Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA
[b] SAIC, EMC/NCEP/NOAA, Camp Springs, MD, USA
[c] Earth System Science Interdisciplinary Center, University of Maryland, College Park, MD, USA
[d] UNESCO-IHE Institute for Water Education, Delft, The Netherlands
[e] National Research Council, Institute for Information Technology, Montreal, Canada

Abstract

This paper introduces a generic theoretical framework for predictive learning, and relates it to data-driven and learning applications in earth and environmental sciences. The issues of data quality, selection of the error function, incorporation of the predictive learning methods into the existing modeling frameworks, expert knowledge, model uncertainty, and other application-domain specific problems are discussed. A brief overview of the papers in the Special Issue is provided, followed by discussion of open issues and directions for future research.

Keywords: Neural networks; Predictive learning; Earth sciences; Environment; Climate; Hydrology

1. Introduction

In this editorial paper, we have attempted to reach the following goals (i) to introduce to practitioners a generic framework of the predictive learning (PL) approach (Section 2); (ii) to introduce a simple classification and a brief review of the PL applications in earth and environmental sciences, and discuss specific issues related to these applications (Section 3); (iii) to briefly overview the papers included in this issue (Section 4); and (iv) to highlight the open issues and future research directions.

2. Framework for predictive learning

The problem of predictive learning (aka inductive learning, machine learning, or learning from examples) can be described in different ways (Mitchell, 1997; Ripley, 1996). In this paper, we adopt the framework of statistical learning (Cherkassky & Mulier, 1998; Friedman, 1994; Vapnik, 1982) shown in Fig. 1.

The setting for predictive learning (PL) involves three components:

- *Generator* of random input vectors $\mathbf{x}$, drawn independently from a fixed (but unknown) probability distribution $P(\mathbf{x})$;
- *System* (or teacher) which returns an output value y for every input vector $\mathbf{x}$ according to the fixed conditional distribution $P(y|\mathbf{x})$, which is also unknown;
- *Learning machine*, which implements a set of approximating functions $f(\mathbf{x},w)$, where w is a set of parameters of an arbitrary nature.

The goal of learning is to select a function (from this set) which approximates best the System's response. This selection is based on the knowledge of finite number (n) of training samples $(\mathbf{x}_i,y_i)$, $(i=1,\ldots,n)$ generated according to (unknown) joint distribution $P(\mathbf{x},y)=P(\mathbf{x})P(y|\mathbf{x})$.

The quality of an approximation produced by the learning machine is measured by the discrepancy or loss $L(y, f(\mathbf{x},\omega))$ between the true output produced by the System and its estimate produced by the learning machine for given input $\mathbf{x}$. By convention, the loss takes on non-negative values, so that large positive values correspond to poor approximation. The expected value of the loss is given by the *prediction risk functional*:

$$R(\omega) = \int L(y, f(\mathbf{x}, \omega))\mathrm{d}P(\mathbf{x}, y) \quad (1)$$

* Corresponding author.
E-mail addresses: cherkass@ece.umn.edu (V. Cherkassky), vladimir.krasnopolsky@noaa.gov (V. Krasnopolsky), d.solomatine@unesco-ihe.org (D.P. Solomatine), julio.valdes@nrc-cnrc.gc.ca (J. Valdes).

0893-6080/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2006.01.001

Learning is the process of finding the function $f(\mathbf{x},\omega_0)$, which minimizes the risk functional (1) over the set of functions supported by the learning machine, using only finite training data (since $P(\mathbf{x},y)$ is unknown). We also point out that the loss function $L(y, f(\mathbf{x},\omega))$ is given a priori based on the problem/application requirements. The prediction risk (1) is unknown, but in practice can be estimated using an independent test set, or via resampling techniques. This formulation (as stated above) is very general and describes many learning problems such as interpolation, regression, classification, and density approximation (Cherkassky & Mulier, 1998; Friedman, 1994; Hastie, Tibshirani, & Friedman, 2001; Vapnik, 1982, 1995).

The problem encountered by the learning machine is to select a function (from the set of functions it supports) that best approximates the System's response. The learning machine is limited to observing finite number (n) examples in order to make this selection. This training data as produced by the generator and system will be independent and identically distributed (iid) according to the joint probability density function (pdf) $p(\mathbf{x},y)$. The finite sample (training data) from this distribution is denoted by:

$$(\mathbf{x}_i, y_i), \quad (i = 1, \ldots, n) \tag{2}$$

With finite data, we cannot expect to find the solution $f(\mathbf{x},\omega_0)$ minimizing prediction risk (1) exactly, so we denote $f(\mathbf{x},\omega^*)$ as the estimate of the optimal solution obtained with finite training data using some learning procedure. It is clear that any learning task (regression, classification, etc.) can be solved by minimizing (1) if the density $p(\mathbf{x},y)$ is known. This means that density estimation is the most general (and hence most difficult) type of learning problem. The problem of learning (estimation) from finite data alone is inherently ill posed. To obtain a useful (unique) solution, the learning process needs to incorporate a priori knowledge in addition to data. For example, such a priori knowledge may be reflected in the set of approximating functions of a learning machine.

Note that a generic learning system shown in Fig. 1 may have two distinct interpretations. Under classical statistical framework, the goal of learning is accurate *identification* of the unknown System, whereas under *predictive learning* (PL) the goal is accurate *imitation* (of a System's output). It should be clear that the goal of system identification is much more demanding than the goal of system imitation. For instance, accurate system identification does not depend on the distribution of input samples; whereas good predictive model is usually conditional upon this (unknown) distribution. Hence,

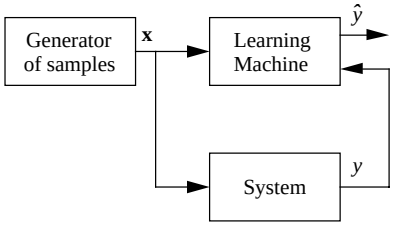

Fig. 1. A learning machine using observations of the system to form an approximation of its output.

an accurate model (in the sense of System's identification) would certainly provide good generalization (in the predictive sense), but the opposite may not be true. The mathematical treatment of system identification leads to the function approximation framework, and to fundamental problems of estimating multivariate functions known as the curse of dimensionality. On the other hand, the goal of accurate system imitation (via minimization of prediction risk) leads to more tractable learning formulations under finite sample settings (Vapnik, 1982, 1995). However, the VC-theoretical approach to PL also requires an appropriate learning problem formulation. This *problem specification* step performs mapping of application-domain requirements onto an appropriate PL formulation, as discussed in Section 3.

Many learning methods are based on the standard (inductive) formulation of the learning problem presented above. For example, a given application is usually formalized as either standard classification or regression problem, even when such standard formulations do not reflect application requirements. Such inductive learning settings assume that:

- the number of future (test) samples is very large, as implied in the expression for risk (1). Moreover, the input ($\mathbf{x}$) values of test samples are unknown during model estimation (training);
- the goal of learning is to model the training data using a single (albeit complex) model;
- the learning machine (in Fig. 1) has a univariate output;
- specific loss functions are used for classification and regression problems.

These assumptions may not hold for many applications. For example, if the input values of the test samples are known (given), then an appropriate goal of learning may be to predict outputs *only* at these points. This leads to the transduction formulation (Vapnik, 1995). Relaxing the assumption about estimating (learning) a single model leads to multiple model estimation formulation (Cherkassky & Ma, 2005). Likewise, it may be possible to relax the assumption about a univariate output under standard supervised learning settings. In many applications, it is necessary to estimate multiple outputs (multivariate functions) of the same input variables. Such methods (for estimating multiple output functions) have been widely used by practitioners, i.e. partial least squares (PLS) regression in chemometrics (Frank & Friedman, 1993). Further, standard loss functions (in classification or regression formulations) may not be appropriate for many applications.

Even though the *problem specification* step cannot be formalized, we suggest several useful guidelines to aid practitioners in the formalization process (Cherkassky, 2001, 2005). The block diagrams for mapping application requirements onto a learning formulation (shown in Fig. 2) advocates the top–down process for specifying three important components of the problem formulation (loss function, input/output variables, and training/test data) based on application needs. In particular, this may include: