Learning with Regularization Networks in Bang

Petra Kudová

Department of Theoretical Computer Science
Institute of Computer Science
Academy of Sciences of the Czech Republic

HPC Europa visitor - EPCC, Edinburgh

Outline

- Project background
  - system Bang
  - my visit
- Regularization Networks
  - learning as function approximation
  - regularization theory
  - learning algorithm
  - model selection
- Experimental results
  - the role of regularization parameter and kernel
  - comparison of different kernel functions
  - prediction of flow rate - data from real life
- Summary and future work

epcc

Bang project http://bang.sf.net

- a project developed at the Institute of Computer Science
- by a group of Phd students under supervison of R. Neruda
- What is Bang? Official short answer:

Bang is a multi-agent system (MAS) intended primarily for experimenting with computational intelligence models. It is a distributed, multiprocess/multi-thread, user-friendly, modular environment allowing for data-driven hybrid modeling with components like artificial neural networks, genetic algorithms, etc. As other MASs, Bang consists of environment and agents that communicate via messages.

Bang project

Motivation

- soft computing, neural networks, genetic algorithms
- environment for experiments with AI methods and easy creation of hybrid models

Goals

- individual methods encapsulated into agents
- agents can communicate and cooperate on a given task
- ideal state: agents are able to autonomously decide which method is suitable for given problem, based on previous knowledge

My visit

Work done during my visit

- porting the system Bang on Lomond machine
- developing the framework for experiments
- experiments with RBF networks and RN networks

Goal of experiments

- comparison of individual learning algorithms
- experiments with RN learning
- setup of explicit parameters
- choice of kernel function

Experimental study of learning algorithms

Why we do experiments?

- there is always gap between theory and practise
- we want to study the real performance and applicability
- verify theoretical results

Experiments

- typically many evaluations of the algorithm
  - wide range of tasks
  - different setup
- results in many independent computations
- straightforward parallelisation

|epcc|

Regularization Networks

Regularization Networks

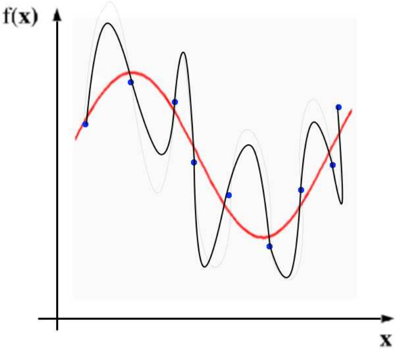
- method for supervised learning
- a family of feedforward neural networks with one hidden layer
- derived from regularization theory
- very good theoretical background

Our focus

- we are interested in their real applicability
- setup of explicit parameters

Learning from examples - problem statement

- **Given:** set of data samples $\{(\vec{x}_i, y_i) \in R^d \times R\}_{i=1}^N$
- **Our goal:** recover the unknown function or find the best estimate of it



Regularization Theory

Empirical Risk Minimization:

- find f that minimizes $H[f] = \sum_{i=1}^{N}(f(\vec{x}_i) - y_i)^2$
- generally ill-posed
- choose one solution according to a priori knowledge (*smoothness, etc.*)

Regularization approach

- add a **stabiliser** $H[f] = \sum_{i=1}^{N}(f(\vec{x}_i) - y_i)^2 + \gamma\Phi[f]$

Derivation of Regularization Network

Stabilizer based on Fourier transform

[Girosi, Jones, Poggio, 1995]

- reflects some knowledge about the target function (usually smoothness, etc.)
- penalize functions that oscillate too much
- stabilizer in a form:

$$\Phi[f] = \int_{R^d} d\vec{s} \frac{|\tilde{f}(\vec{s})|^2}{\tilde{G}(\vec{s})}$$

$\tilde{f}$	Fourier transform of f	$\tilde{G}(\vec{s}) \to 0$ for $\|\|s\|\| \to \infty$
$\tilde{G}$	positive function	$1/\tilde{G}$ high-pass filter

Derivation of Regularization Network

Form of the solution

- for a wide class of stabilizers (G positive semi-definite) the solution has a form

$$f(x) = \sum_{i=1}^{m} w_i G(\vec{x} - \vec{x}_i)$$

- where weights w_i satisfy

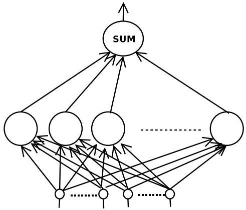
$$(\gamma I + G)\vec{w} = \vec{y}$$

- represented by feed-forward neural network with one hidden layer

Derivation of Regularization Network

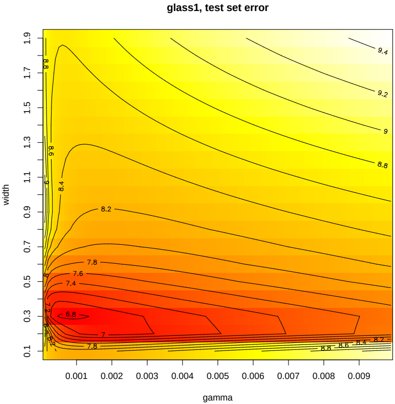
Regularization network

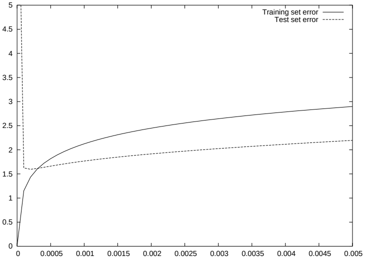
$$f(x) = \sum_{i=1}^{m} w_i G(\vec{x} - \vec{x}_i)$$



- function G called basis or kernel function
- choice of G represents our knowledge or assumption about the problem
- choice of G is crucial for the generalization performance of the network

The role of γ and kernel function





Model selection

Parameters of the basic algorithm

- kernel type
- kernel parameter(s) (i.e. width for Gaussian)
- regularization parameter γ

How we estimate these parameters?

- kernel type by user
- kernel parameter and regularization parameter by grid search and cross-validation
- speed-up techniques: grid refining

Experiments

Data

- Lenna image - approximation
- benchmark data sets - Proben1 data repository
- prediction of flow rate on rivers Sázava and Ploučnice

Methodology

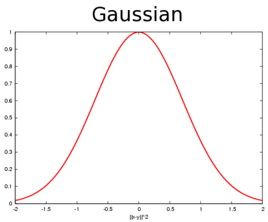
- separate data for training and testing
- find suitable kernel function and γ on training set by cross-validation
- learn on training set (estimation of weights w)
- evaluate error on testing set - generalization ability

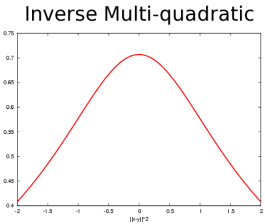
Lenna - approximation

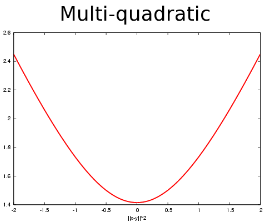
Columns: 0.0, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}

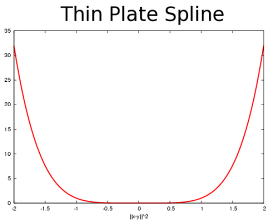
Rows: 0.5, 1.0, 1.5, 2.0

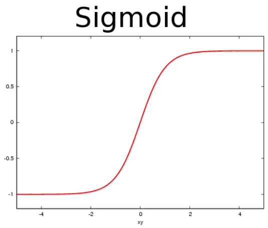
Proben1 - comparison of kernel functions



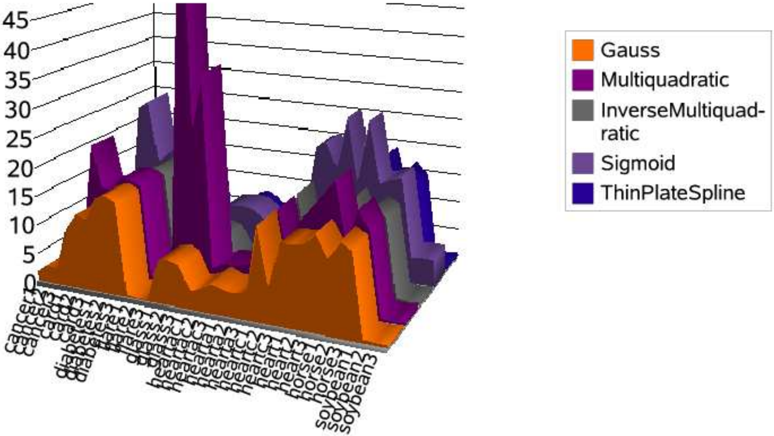






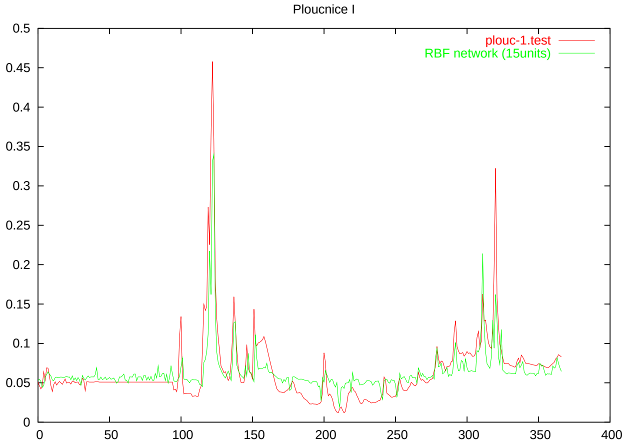


Proben1 - comparison of kernel functions



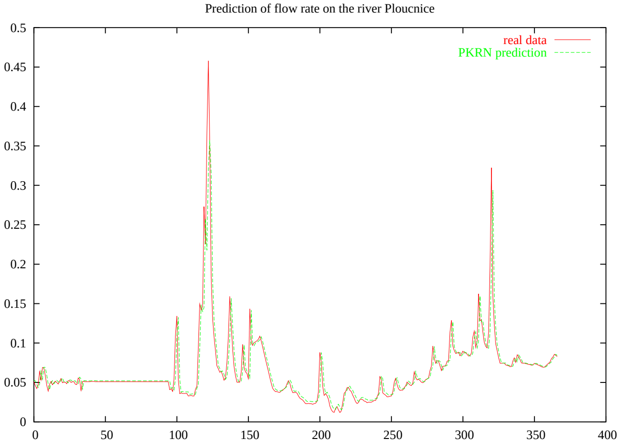
Prediction of flow rate on river Ploučnice

Prediction by RBF network



Prediction of flow rate on river Ploučnice

Prediction by Product Kernels



Summary and future work

Summary

- Bang project was briefly introduced
- RN networks described
- selected results of experiments shown

Work in progress and future work

- automated choice of kernel function
- composite type of kernels

Thank you! Questions?