Learning with Regularization Networks

Petra Kudová

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

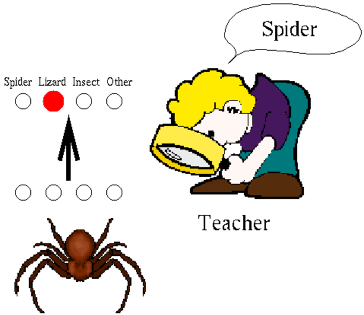
Outline

- Introduction
  - supervised learning
- Regularization Networks
  - regularization theory, RN learning algorithm
  - composite kernels
- Generalized Regularization Networks
  - RBF networks
- Flow rate prediction
- Summary and Future Work

Supervised Learning

Learning

- given set of data samples
- find underlying trend, description of data



Supervised Learning

- data – input-output patterns
- create model representing IO mapping
- classification, regression, prediction, etc.

Regularization Networks

Regularization Networks

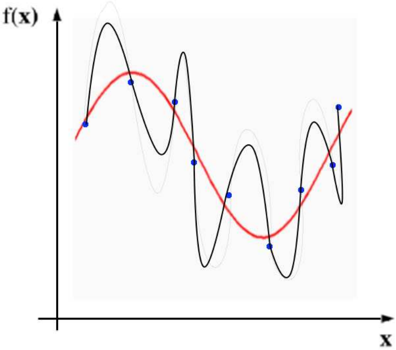
- method for supervised learning
- a family of feed-forward neural networks with one hidden layer
- derived from regularization theory
- very good theoretical background

Our Focus

- we are interested in their real applicability
- setup of explicit parameters – choice of kernel function

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 prior 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

- for a wide class of stabilizers the solution of

$$\min_{f \in \mathcal{H}} H[f]; \qquad \text{where } H[f] = \sum_{i=1}^{N} (f(\vec{x}_i) - y_i)^2 + \gamma \Phi[f]$$

exists and is unique

- many proofs
  - *Girossi, Poggio, Jones (1995)* – using stabilizers based on Fourier transform
  - *Smale, Poggio (2003)* – using RKHS
  - others

Derivation using RKHS

- Data set: $\{(\vec{x}_i, y_i) \in R^d \times R\}_{i=1}^N$
- choose a symmetric, positive-definite kernel $K = K(\vec{x}_1, \vec{x}_2)$
- let $\mathcal{H}_K$ be the RKHS defined by K
- define the stabiliser by the norm $|| \cdot ||_K$ in $\mathcal{H}_K$

$$H[f] = \sum_{i=1}^N (y_i - f(\vec{x}_i))^2 + \gamma ||f||_K^2$$

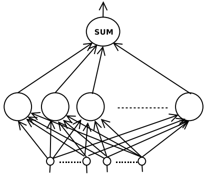
- minimise $H[f]$ over $\mathcal{H}_K$ $\longrightarrow$ solution:

$$f(\vec{x}) = \sum_{i=1}^N w_i K_{\vec{x}_i}(\vec{x}) \qquad (\gamma I + K)\vec{w} = \vec{y}$$

Regularization Network

Network Architecture

$$f(x) = \sum_{i=1}^{N} w_i K(\vec{x}, \vec{x}_i)$$



- function K called basis or kernel function

Basic Algorithm

1. set the centers of kernel functions to the data points
2. compute the output weights by solving linear system

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

Model Selection

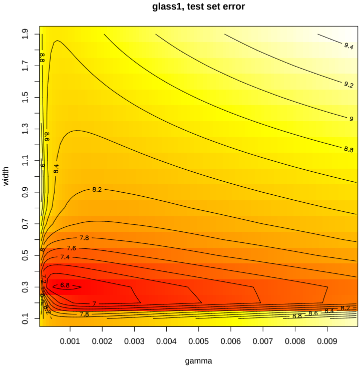
Parameters of the Basic Algorithm

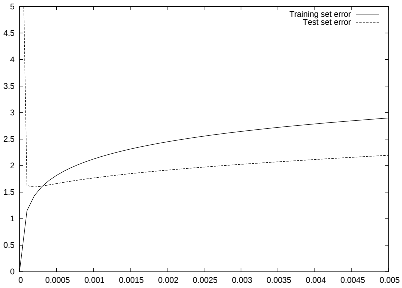
- kernel type
- kernel parameter(s) (i.e. width for Gaussian)
- regularization parameter γ

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

Choice of Regularization Parameter and Kernel

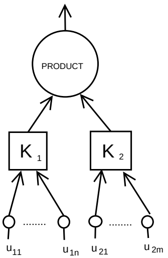


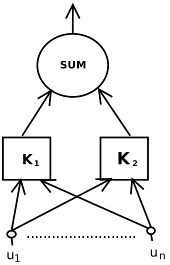


Composite kernels

Product and Sum Kernels

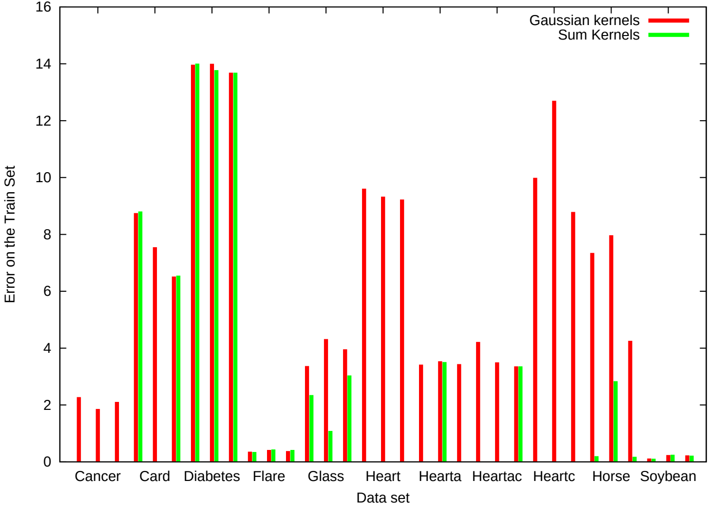
- choice of kernels depands on data, attributes types
- sometime data are not homogenous
- *composite kernels* – product and sum kernels may better reflect the character of data (joint work with T. Šámalová)
- based on Aronszajn theoretical results





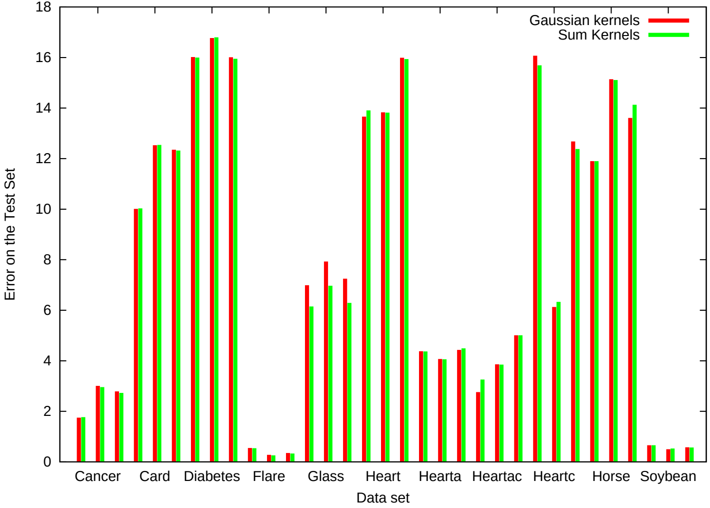
Sum versus Gaussian Kernels

The error on the training set



Sum versus Gaussian Kernels

The error on the testing set



Generalized Regularization Networks

Generalized RN

- less hidden units (kernel functions) than training data points
- centers of kernels distributed using various heuristics (i.e. simple clustering)
- hidden kernel units may have additional parameters

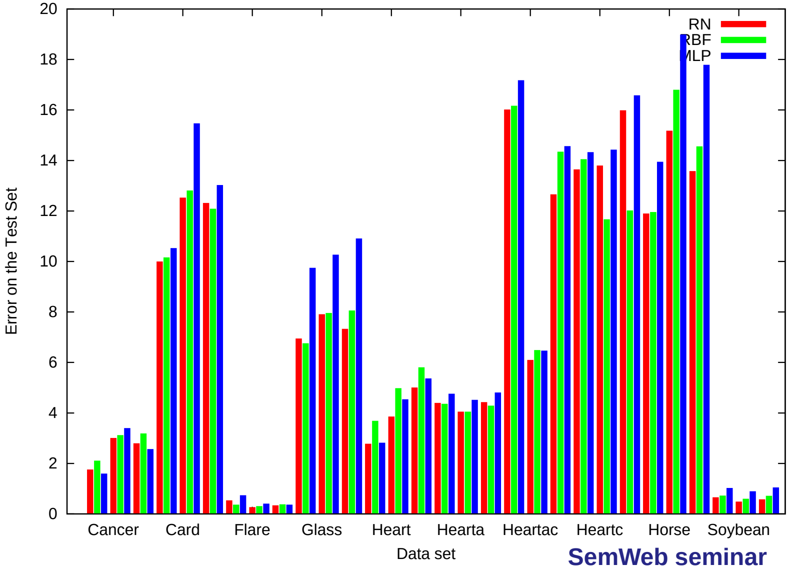
RBF networks

- one class of generalized RN
- derived using radial stabilizers
- wide range of learning algorithms

RN versus RBF networks

Regularization Networks	RBF networks
architecture	
good theoretical background, optimal solution	approximate solution (lower number of hidden units)
learning	
solving linear systems by numerical algorithms	algorithms for optimisation, heuristics
network complexity	
number of parameters depends on the training set size	does not depend on the train. set size, but units have more parameters
parameters (γ, width)	parameter h

Comparison of RN and RBF on Proben1 repository



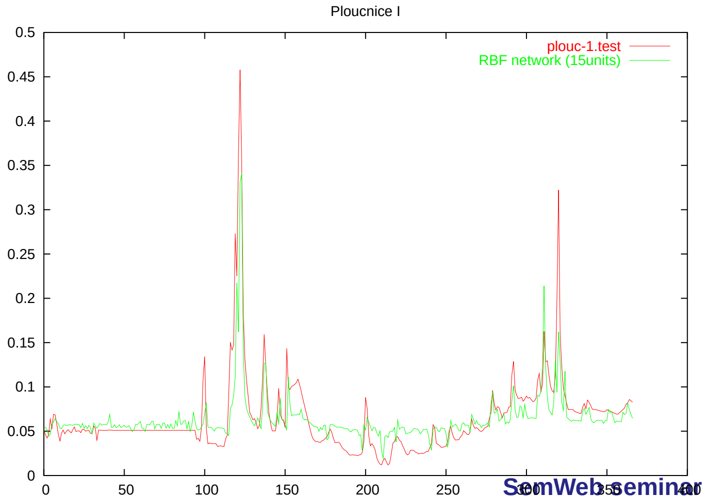
Prediction of flow rate

- prediction of the flow rate on the Ploučnice in North Bohemia, from origin (southwest part of the Ještěd hill) to the town Mimoň

- time series containing daily flow and rainfall values
- prediction of the current flow rate based on information from the previous one or two days
- 1000 training samples, 367 testing samples

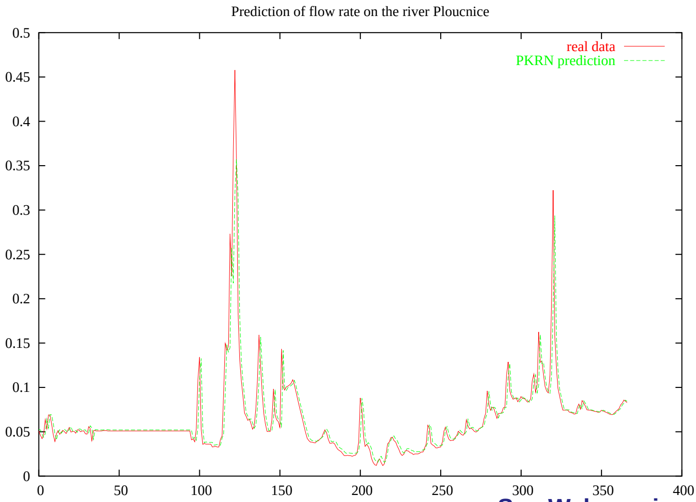
Prediction of flow rate

Prediction by RBF network



Prediction of flow rate

Prediction by Product Kernels



Summary and Future Work

Summary

- learning with RN networks
- composite kernels
- generalized regularization networks
- flow rate prediction

Work in Progress and Future Work

- composite types of kernels
- kernel functions for other data types (categorical data, etc.)

Thank you! Questions?