LEARNING WITH KERNEL BASED REGULARIZATION NETWORKS

P. Kudová

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

ITAT 2005, Roháče, Slovakia
Outline

- Introduction
  - Learning from examples
- Regularization Networks
  - regularization theory
  - learning algorithm
- Estimation of regularization parameter and kernel type
  - crossvalidation
  - adaptive grid search
  - genetic parameter search
- Experiments
  - composite kernels
  - comparison of different types of kernels
Learning from examples - problem statement

- **Given:** set of data samples \( \{(\vec{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{R}\}_{i=1}^N \)

- **Our goal:** recover the unknown function or find the best estimate of it
Examples of learning techniques

- Artificial neural networks
  - biological motivations
  - black box approach
  - multilayer perceptrons, RBF networks, etc.

- Decision trees, Support vector machines, etc.

- Approximation theory
  - function approximation
  - regularization theory
Regularization Theory

Empirical Risk Minimization:

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

Regularization approach

- add a stabiliser $H[f] = \frac{1}{N} \sum_{i=1}^{N} (f(\tilde{x}_i) - y_i)^2 + \gamma \Phi[f]$
Reproducing Kernel Hilbert Space

Definition and properties

- RKHS is a Hilbert space of functions defined over $\Omega \subset \mathbb{R}^d$ with the property that for each $x \in \Omega$ the evaluation functional on $\mathcal{H}$ given by $\mathcal{F}_x : f \rightarrow f(x)$ is bounded. (Aronszajn, 1950)

- This implies existence of positive definite symmetric function $K : \Omega \times \Omega \rightarrow \mathbb{R}$ (kernel function) such that

$$\mathcal{H} = \mathcal{H}_K = \text{comp}\{\sum_{i=1}^{n} a_i K_{x_i}; \ x_i \in \Omega, a_i \in \mathbb{R}\},$$

where $\text{comp}$ means completion of the set.
RKHS and learning

- Data set: \( \{ (\vec{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{R} \}_{i=1}^N \)
- 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 \( \mathcal{H}_K \)

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

minimise \( H[f] \) over \( \mathcal{H}_K \) \( \longrightarrow \) solution:

\[
f(\vec{x}) = \sum_{i=1}^{N} c_i K_{\vec{x}_i}(\vec{x}) \quad \quad (N\gamma I + K)\vec{c} = \vec{y}
\]
RN learning algorithm

- algorithm is quite simple, reduces to solving a linear system

\[(N\gamma I + K)\tilde{c} = \tilde{y}\]

- \(N\gamma I + K\) strictly positive \(\rightarrow\) system is \textbf{well-posed}

- Is it also \textbf{well-conditioned}?

- For large \(\gamma\) \(\rightarrow\) dominant diagonal \(\rightarrow\) good

- choice of \(\gamma\) is not free

- choice of kernel and its parameters

- choice of \(\gamma\) and kernel type is crucial for the performance of the algorithm
Choice of $\gamma$ and type of kernel

**Glass1, test set error**

- **X-axis**: Gamma values ranging from 0.001 to 0.009.
- **Y-axis**: Width values ranging from 0.1 to 1.9.

- **Graph**: Shows the variation of test set error with different gamma and width values.

**Training set error** vs **Test set error**

- **X-axis**: Gamma values ranging from 0.0005 to 0.005.
- **Y-axis**: Error values ranging from 0 to 5.

- **Graph**: Compares training set error and test set error for different gamma values.
Model selection

Parameters of the basic algorithm

- kernel type
- kernel parameter(s) (i.e. width for Gaussian)
- regularization parameter $\gamma$

How we estimate these parameters?

- kernel type by user
- kernel’s parameters and regularization parameter - search for parameters with the lowest cross-validation error
- grid search, genetic search
Parameter search - Adaptive Grid

Move grid

- Adaptive Grid
- Model selection
- Experiments
- Conclusion

ITAT, 20. - 24. 9. 2005
Parameter search - Adaptive Grid

Refine grid

![Diagram showing adaptation of grid parameters]

- **width**
- **reg. parameter**
- **new grid**
- **min**
Parameter search - Adaptive Grid

![Graph showing parameter search with adaptive grid]
Parameter search - Adaptive Grid
Parameter search - Adaptive Grid
Parameter search - Adaptive Grid
Parameter search - Adaptive Grid
Genetic parameter search

Adaptive grid

- time consuming, many evaluations
- danger of overfitting (with respect to a particular partitioning to folds)

Genetic algorithms

- applying simple genetic algorithm
- fitness function - the lower the cross-validation error the higher the fitness
- random partitioning in cross-validation
- lazy evaluations
Genetic algorithms (GA)
GA operators for parameter search

Individual

Individual used for search including kernel type:

<table>
<thead>
<tr>
<th>type of kernel</th>
<th>kernel parameters</th>
<th>reg. parameter</th>
</tr>
</thead>
</table>

Individual used for Gaussian kernels:

<table>
<thead>
<tr>
<th>width</th>
<th>reg. parameter</th>
</tr>
</thead>
</table>

Crossover

\[
\begin{align*}
    b_1 & \quad \text{gamma}_1 \\
    b_2 & \quad \text{gamma}_2 \\
    \text{crossover} & \quad \text{a}_1 \times \text{b}_1 + (1-\text{a}_1) \times \text{b}_2 \quad \text{c}_1 \times \text{gamma}_1 + (1-\text{c}_1) \times \text{gamma}_2 \\
    & \quad \text{a}_2 \times \text{b}_1 + (1-\text{a}_2) \times \text{b}_2 \quad \text{c}_2 \times \text{gamma}_1 + (1-\text{c}_2) \times \text{gamma}_2
\end{align*}
\]
Genetic algorithm - example

"41_genetics.log"  
"41_genetics.log" using 1:4
Experiments

- separate data for training and testing
- find parameters using training data set
- run RN learning using winning parameters
- evaluate error on the testing data set

\[ E = 100 \frac{1}{N} \sum_{i=1}^{N} ||\tilde{x}^i - f(\tilde{x}^i)||^2 \]

- Proben1 data repository
- both approximation and classification tasks
- each task in 3 different partitioning
choice of kernels depends 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)
Comparison of Gaussian, Sum and Product kernel

Data set: Cancer, Card, Diabetes, Flare, Glass, Heart, Hearta, Heartac, Heartc, Horse, Soybean

Error on the training set

Gaussian kernels, Sum kernels, Product kernels
Comparison of Gaussian, Sum and Product kernel

Error on the training set

Data set: Cancer, Card, Diabetes, Flare, Glass, Heart, Hearta, Heartac, Heartc, Horse, Soybean

Kernels: Gaussian, Sum, Product

ITAT, 20. - 24. 9. 2005
Winning sum unit

![Graph showing winning sum kernel and winning simple kernel for cancer1]
Conclusion

Summary

- RN learning algorithm
- methods for parameter search
- comparison of Gaussian, Product and Sum kernels

Work in progress & future work

- include kernel type in parameter search