TKK Reports in Information and Computer Science

Espoo 2010 TKK-ICS-R33

# FAST MISSING VALUE IMPUTATION USING ENSEMBLE OF SOMS

Antti Sorjamaa and Amaury Lendasse

Aalto University
School of Science
and Technology

TKK Reports in Information and Computer Science

Espoo 2010 TKK-ICS-R33

# FAST MISSING VALUE IMPUTATION USING ENSEMBLE OF SOMS

**Antti Sorjamaa and Amaury Lendasse**

Aalto University School of Science and Technology
Faculty of Information and Natural Sciences
Department of Information and Computer Science

Aalto-yliopiston teknillinen korkeakoulu
Informaatio- ja luonnontieteiden tiedekunta
Tietojenkäsittelytieteen laitos

Distribution:
Aalto University School of Science and Technology
Faculty of Information and Natural Sciences
Department of Information and Computer Science
PO Box 15400
FI-00076 AALTO
FINLAND
URL: http://ics.tkk.fi
Tel. +358 9 470 01
Fax +358 9 470 23369
E-mail: series@ics.tkk.fi

ISBN 978-952-60-3247-4 (Print)
ISBN 978-952-60-3248-1 (Online)
ISSN 1797-5034 (Print)
ISSN 1797-5042 (Online)
URL: http://lib.tkk.fi/Reports/2010/isbn9789526032481.pdf

TKK ICS
Espoo 2010

**ABSTRACT:** This report presents a methodology for missing value imputation. The methodology is based on an ensemble of Self-Organizing Maps (SOM), which is weighted using Nonnegative Least Squares algorithm. Instead of a need for lengthy validation procedure as when using single SOMs, the ensemble proceeds straight into final model building. Therefore, the methodology has very low computational time while retaining the accuracy. The performance is compared to other state-of-the-art methodologies using two real world databases from different fields.

**KEYWORDS:** Self-Organizing Maps, Ensemble of SOMs, Missing Value Imputation, Nonnegative Least Squares, Fast Computation

# CONTENTS

# Chapter 1

# Introduction

The presence of missing values in the underlying time series is a recurrent problem when dealing with databases. Number of methods have been developed to solve the problem and fill the missing values.

In this paper, we focus on Self-Organizing Maps [6] (SOM), which aim to ideally group homogeneous individuals, highlighting a neighborhood structure between classes in a chosen lattice. The SOM algorithm is based on unsupervised learning principle where the training is entirely stochastic, data-driven. No information about the input data is required. Recent approaches propose to take advantage of the homogeneity of the underlying classes for data completion purposes [11]. Furthermore, the SOM algorithm allows projection of high-dimensional data to a low-dimensional grid. Through this projection and focusing on its property of topology preservation, SOM allows nonlinear interpolation for missing values.

But how to find optimal SOM size and shape? One of the typical machine learning paradigms is about finding the model that best fits the given data, in terms of test or validation. Searching for such a model can be very time consuming: finding the model class that best suits the type of data, optimizing the possible hyper-parameters, and finally training the model once all details of the model structure have been selected. This procedure can lead to a rather good model, which fits the data and avoids the pitfalls of overfitting.

On the other hand, creating an ensemble of less good models might achieve better performance, while alleviating the problem of extensive validation procedure. Even faster model building is achieved through parallel computation, which is easy to implement when several different models are built.

The goal is then to weight each model in the ensemble so that the overall output has the best possible performance. Several ensemble techniques have been proposed, out of which two kinds can be distinguished [4]: the variable weights approach and the average ones. Traditionally, average weights ensemble techniques are used by simply taking an average of all the built models. While this obviously has the advantage of having immediately the weights of all models, it yields suboptimal results. The variable weights ensemble techniques try to optimize the weight of each model in the ensemble according to a criterion. Techniques such as the Genetic Algorithm have been recently used for such optimization [12] but they are very time consuming.

This paper describes a method, which combines several SOMs in order to enhance the accuracy of the nonlinear interpolation. The ensemble weights are obtained with a classical constrained linear solution, the Nonnegative Least Squares, and it improves the accuracy of the imputation as well as speeds up the process by removing the need for validation.

The following section presents the SOM imputation methodology and Section 3 presents the Ensemble of SOMs. Section 4 compares the performance of the presented methodology against the state-of-the-art imputation methodologies using two real world databases. Finally, conclusions and further work are derived and presented in Section 5.

# Chapter 2

# Imputation using SOM

The SOM algorithm is based on an unsupervised learning principle, where training is entirely data-driven and no information about the input data is required [6]. Here we use a 2-dimensional network, composed of $c$ units (or code vectors) shaped as a square *lattice*. Each unit of a network has as many weights as the length $T$ of the learning data samples, $\mathbf{x}_n$, $n = 1, 2, ..., N$. All units of a network can be collected to a weight matrix $\mathbf{m}\left(t\right) = \left[\mathbf{m}_1\left(t\right), \mathbf{m}_2\left(t\right), ..., \mathbf{m}_c\left(t\right)\right]$ where $\mathbf{m}_i\left(t\right)$ is the $T$-dimensional weight vector of the unit $i$ at time $t$ and $t$ represents the steps of the learning process. Each unit is connected to its neighboring units through a neighborhood function $\lambda(\mathbf{m}_i, \mathbf{m}_j, t)$, which defines the shape and the size of the neighborhood at time $t$. The neighborhood can be constant through the entire learning process or it can change in the course of learning.

The learning starts by initializing the network node weights randomly. Then, for a randomly selected sample $\mathbf{x}_{t+1}$, we calculate the Best Matching Unit (BMU), which is the neuron whose weights are closest to the sample. The BMU calculation is defined as

$$\mathbf{m}_{BMU(\mathbf{x}_{t+1})} = \arg \min_{\mathbf{m}_i, i \in I} \left\{ \left\| \mathbf{x_{t+1}} - \mathbf{m}_i\left(t\right) \right\| \right\}, \tag{2.1}$$

where $I = [1, 2, ..., c]$ is the set of network node indices, the $BMU$ denotes the index of the best matching node and $\|.\|$ is a standard Euclidean norm.

If the randomly selected sample includes missing values, the BMU cannot be solved outright. Instead, an adapted SOM algorithm, proposed by Cottrell and Letrémy [3], is used. The randomly drawn sample $\mathbf{x_{t+1}}$ having missing value(s) is split into two subsets $\mathbf{x}_{t+1}^T = NM_{\mathbf{x_{t+1}}} \cup M_{\mathbf{x_{t+1}}}$, where $NM_{\mathbf{x_{t+1}}}$ is the subset where the values of $\mathbf{x_{t+1}}$ are not missing and $M_{\mathbf{x_{t+1}}}$ is the subset, where the values of $\mathbf{x_{t+1}}$ are missing. We define a norm on the subset $NM_{\mathbf{x_{t+1}}}$ as

$$\left\| \mathbf{x_{t+1}} - \mathbf{m}_i\left(t\right) \right\|_{NM_{\mathbf{x_{t+1}}}} = \sum_{k \in NM_{\mathbf{x_{t+1}}}} \left(\mathbf{x}_{t+1,k} - \mathbf{m}_{i,k}(t)\right)^2, \tag{2.2}$$

where $\mathbf{x}_{t+1,k}$ for $k = [1, ..., T]$ denotes the $k^{th}$ value of the chosen vector and $\mathbf{m}_{i,k}(t)$ for $k = [1, ..., T]$ and for $i = [1, ..., c]$ is the $k^{th}$ value of the $i^{th}$ code vector.

Then the BMU is calculated with

$$\mathbf{m}_{BMU(\mathbf{x}_{t+1})} = \arg \min_{\mathbf{m}_i, i \in I} \left\{ \left\| \mathbf{x_{t+1}} - \mathbf{m}_i \left( t \right) \right\|_{NM_{\mathbf{x_{t+1}}}} \right\}. \tag{2.3}$$

When the BMU is found the network weights are updated as

$$\mathbf{m}_i \left( t+1 \right) = \mathbf{m}_i \left( t \right) - \varepsilon(t) \lambda \left( \mathbf{m}_{BMU(\mathbf{x}_{t+1})}, \mathbf{m}_i, t \right) \left[ \mathbf{m}_i \left( t \right) - \mathbf{x_{t+1}} \right], \forall i \in I, \tag{2.4}$$

where $\varepsilon(t)$ is the adaptation gain parameter, which is $]0,1[$-valued, decreasing gradually with time. The number of neurons taken into account during the weight update depends on the neighborhood function $\lambda(\mathbf{m}_i, \mathbf{m}_j, t)$. The number of neurons, which need the weight update, usually decreases with time.

After the weight update the next sample is randomly drawn from the data matrix and the procedure is started again by finding the BMU of the sample. The learning procedure is stopped when the SOM algorithm has converged.

Once the SOM algorithm has converged, we obtain some clusters containing our data. Cottrell and Letrémy proposed to fill the missing values of the dataset by the coordinates of the code vectors of each BMU as natural first candidates for the missing value completion:

$$\pi_{(M_{\mathbf{x}})} \left( \mathbf{x} \right) = \pi_{(M_{\mathbf{x}})} \left( \mathbf{m}_{BMU(\mathbf{x})} \right), \tag{2.5}$$

where $\pi_{(M_{\mathbf{x}})}(.)$ replaces the missing values $M_{\mathbf{x}}$ of sample $\mathbf{x}$ with the corresponding values of the BMU of the sample. The replacement is done for every data sample and then the SOM has finished filling the missing values in the data.

The procedure is summarized in Table 2.1. There is a toolbox available for performing the SOM algorithm in [1].

Table 2.1: Summary of the SOM algorithm for finding the missing values.

1. SOM node weights are initialized randomly
2. SOM learning process begins
    - (a) Input $\mathbf{x}$ is drawn from the learning data set $\mathbf{X}$
        - i. If $\mathbf{x}$ does not contain missing values, BMU is found according to Equation 2.1
        - ii. If $\mathbf{x}$ contains missing values, BMU is found according to Equation 2.3
    - (b) Neuron weights are updated according to Equation 2.5
3. Once the learning process is done, for each observation containing missing values, the weights of the BMU of the observation are substituted for the missing values

# Chapter 3

# Ensemble of SOMs

The Ensemble of SOMs methodology is summarized in Figure 3.1.

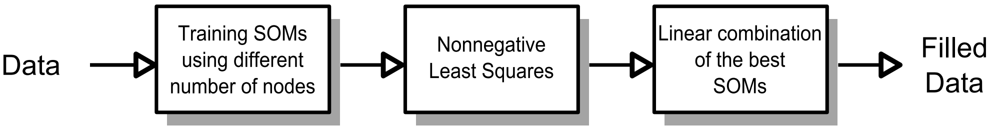
Data
Training SOMs using different number of nodes
Nonnegative Least Squares
Linear combination of the best SOMs
Filled Data

Figure 3.1: Ensemble of SOMs summarized.

The core of the methodology is the Self-Organizing Map (SOM). Several SOMs are trained using different number of nodes and the imputation results of the SOMs are linearly combined. The linear coefficients are computed using Nonnegative Least Squares (NNLS) algorithm.

Each SOM map $M_i$ has different number of nodes and the given imputation estimates for the missing values in the database are also different. For each missing value $y$, every SOM in the combination is giving an estimation $\hat{y}_i$ and the final estimation of the missing value $\hat{y}$ is the linear combination of the individual SOM estimates. The combination procedure is shown in Figure 3.2.

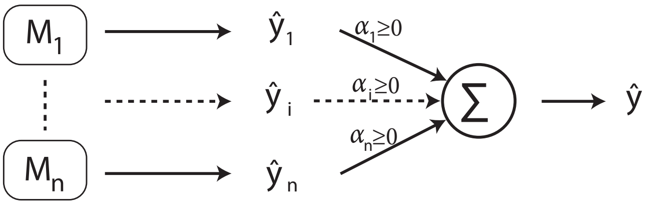

Figure 3.2: Illustrative scheme of the combination of SOMs.

Assuming that each SOM is unbiased, the combination can be made unbiased by having $\sum \alpha_i = 1$. Negative weights have no physical meaning and would lead to overfitting, hence the weights have to be restricted to be nonnegative [7, 9].

For the determination of the weights $\alpha_i$, a classical constrained optimization method called Non-Negative constrained Least-Squares (NNLS) algorithm [8] is used to compute the solution.

For the computation, a small set of the data is removed and used as a *calibration set*. The size of the calibration set has to be selected with respect to the number of missing values in the database. Selecting too many calibration points in a database having large percentage of missing values already results to inaccuracy due to the lack of data. On the other hand, selecting too few calibration points decreases the accuracy of computing the weights for the SOM estimates. In any case, number of calibration points should be larger than the number of SOMs in the ensemble.

When the calibration set has been estimated by the SOMs, the coefficients $\alpha_i$ are solved from the constrained linear system, shown in Equation (3.1), using the NNLS algorithm.

$$\arg\min_{\boldsymbol{\alpha}} \left\| \boldsymbol{y}^{\mathrm{Cal}} - \sum_{i=1}^{n} \alpha_i \hat{\boldsymbol{y}}_i^{\mathrm{Cal}} \right\|^2 \quad \text{s.t.} \;\; \alpha_i \geq 0\,. \tag{3.1}$$

After using the NNLS algorithm, the Ensemble of SOMs can be used to fill the missing values on the whole database.

The way of combining the SOM estimates of missing values removes the need for lengthy and time consuming Cross-Validation procedure needed in traditional SOM imputation. Each SOM size needs to be trained only once and the most accurate ones are combined using the NNLS and the calibration set. Furthermore, the linear combination by the NNLS is known to converge in $\frac{1}{2}n$ steps [8] and the result is notably more accurate than any individual SOM map used in the combination.

# Chapter 4

# Experiments

In the following, the datasets used in the experiments are introduced. The datasets are very diverse and coming from very different fields, namely, corporate finance and climatology. Both of the datasets are tough to fill, due to their nonlinear nature and inherently present missing values.

The Ensemble of SOMs methodology is compared against other imputation methodologies, which are presented briefly after the datasets.

## 4.1 DATASETS

### 4.1.1 Corporate Finance

The corporate finance data collects information about companies and their performance[1]. The information is completely numerical and it inherently includes 14 percent of missing values.

The source of the data is Thomson One Banker and it includes almost 6000 French and British companies. Each company is represented by 45 yearly key numbers from years 1999 to 2006, including three binary variables for the operative field. All companies are either registered in Paris Stock Exchange or London Stock Exchange and most of the companies are medium sized (51-500 employees) or large (more than 500 employees).

In the 45 key numbers, some characteristics such as assets, current assets, total debt or total equity are taken into account. The objective in this dataset is to build indicators able to explain the variable long term debt (Y) (i.e. long term debt/total debt). 7 variables are built. Each variable is an indicator to explain Y. The main indicators are Market value of shares/Book value of shares, variation of sales, Altman's score, size, corporate performance (EBITDA/Total assets), industrial sector and the characteristics of the legal system (creditor oriented or common law system for UK firms and debtor oriented or civil law système for French firms).

[1] Since the data included in this dataset is confidential, it is not publicly available.

### 4.1.2 Tanganyika Lake Temperature

The third dataset, Tanganyika Lake Surface Temperature dataset[2], is the largest one of the three datasets used in the experiments. The Tanganyika lake lies in the African Rift area and it is over 670 kilometers long with an average width of about 50 kilometers. The measurements are obtained from the thermal infrared bands of the MODIS sensors of the satellite covering the lake with a spatial resolution of one kilometer.

The satellite has measured the lake a total of 666 times between years 2002 and 2006. The measuring frequency of the satellite is not constant during the five year period, instead it varies from one to 33 days. On average we have one image every 2,5 days.

The spatial resolution gives us more than 32 000 daily measurement points in one image. The amount of missing values in each image varies from 100 percent to four percent, meaning that some images have no measurement values and some have only four percent of the data missing. Finally, the whole dataset has over 63 percent of the data missing.

Because of the huge size of the dataset, it is divided into slices. Each measurement image of the lake is cut to ten pieces in north-south direction. This is done in order to take into account the change in the dynamics of the long lake and to make the filling more local. Moreover, the percentage of missing values is on average greater in the northern part of the lake whereas the middle and the southern parts have more measurements present.

As an example slice, the most southern part of the lake is shown in Figure 4.1 and used as an example slice in the following.

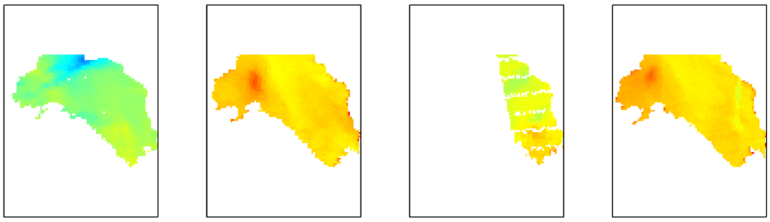

Figure 4.1: Southern slice of the Tanganyika Lake dataset, days from 182 to 185 from 2005.

Because of the large number of missing data in the database, each day with more than 90 percent of the data missing from the slice is removed from the dataset before the learning phase. This is done for each slice individually. For the southernmost slice of the Tanganyika Lake, it leaves us 390 days with each day containing a total of 2947 measurement points with a total of 27 percent of the data missing on average.

[2] Tanganyika Lake dataset, MODIS Data, a courtesy of Yves Cornet and the University of Liege, Belgium. The data comes from an RS dataset produced in the framework of the CKIMFISH project.

## 4.2 COMPARED METHODOLOGIES

### 4.2.1 EOF

Empirical Orthogonal Functions (EOF) methodology is based on Singular Value Decomposition (SVD) algorithm. In a nutshell, after initialization of the missing values, the SVD is used to compute the singular vectors and values, which are used to reconstruct the dataset. The reconstructed data replaces the missing values in the original set and the procedure is repeated several times. After the missing values are not changing anymore, the methodology terminates.

One needs to select the number of singular values and vectors to be used in the reconstruction. This is achieved using Cross-Validation method and trying out several possible values. For more information about EOF, see [2].

### 4.2.2 EOF Pruning

The EOF Pruning[3] improves the original EOF methodology. Whereas the EOF selects a certain number of largest singular values and vectors, the EOF Pruning selects them in a non-continuous fashion. Otherwise the procedure is the same than with the original EOF, but the EOF Pruning requires only a few rounds of computation using SVD and most of the computation time is spent on selecting the singular values and vectors. For more information on EOF Pruning, see [10].

### 4.2.3 Probabilistic PCA

Probabilistic Principal Component Analysis (PPCA) extends the standard PCA by using a probabilistic formulation, which enables the use of data with missing values. In the experiments, we use freely available Matlab toolbox[4] with Variational Bayesian algorithm. For more information on PPCA and the toolbox, see [5]

## 4.3 RESULTS

Before the filling process is started, we need to remove the test set from the data. Test set is removed in order to estimate the accuracy of the methodologies. For SOM, EOF, EOF Pruning and PPCA also validation sets are selected. For Ensemble of SOMs validation sets are not necessary, but still calibration set is needed. The calibration sets are selected in a similar way than validation sets for other methods.

The validation sets and the calibration set are not identical, but each test set is identical in order to be able to compare the methodologies reliably. The whole selection procedure starting from the selection of test sets is completely

[3] There is a toolbox available for performing the EOF Pruning. It can be downloaded from http://www.cis.hut.fi/projects/eiml/research/downloads/someof-toolbox

[4] The PPCA toolbox can be downloaded from
http://www.cis.hut.fi/alexilin/software/

repeated 10 times in order to have robust approximation of the generalization error for each methodology. The selected sets in each repetition are not overlapping with each other. All results are calculated as an average of the 10 repetitions.

In the experiments, when combining the SOM maps, the number of SOM maps varied. Each SOM has different amount of nodes aligned into a two-dimensional lattice using hexagonal neighborhood. The sizes of the SOMs were defined as described in [6] and in [1] in *SOM algorithm implementation in SOM Toolbox*. All SOMs were trained using the default settings.

The results regarding the finance dataset are presented in Table 4.1.

Table 4.1: The results of all methods using Finance dataset.

| **Finance** | Val MSE | Test MSE | Computational Time |
|---|---|---|---|
| EOF | 0.680 | 0.707 | 1.42 hours |
| EOF Pruning | 0.305 | 0.318 | 18.6 minutes |
| SOM | 0.452 | 0.458 | 38.40 minutes |
| Ensemble of SOMs | | 0.424 | 5.9 minutes |
| PPCA | 0.639 | 0.755 | 18.35 hours |

From Table 4.1, we can see that the best methodology, in terms of the accuracy, is EOF Pruning. Even though the Ensemble of SOMs is not as accurate as the EOF Pruning, it is the second most accurate and faster than all other methods.

However, what is surprising with this Finance dataset, is that the test errors are all higher than validation errors. This is counterintuitive from what we have learned when dealing with databases with missing values. This warrants more research into the data to detect why it behaves differently.

Table 4.2 shows the results with the Tanganyika dataset.

Table 4.2: The results of all methods using Tanganyika dataset.

| | Val MSE | Test MSE | Computational Time |
|---|---|---|---|
| EOF | 0.0553 | 0.0595 | 30.3 hours |
| EOF Pruning | 0.0442 | 0.0426 | 23.3 hours |
| SOM | 0.0393 | 0.0379 | 3.14 hours |
| Ensemble of SOMs | | 0.0280 | 39.5 minutes |
| PPCA | 0.0700 | 0.0818 | > 8 days |

From Table 4.2, we can see that according to the test error, the Ensemble of SOMs is the best methodology according to the accuracy. In this case, the EOF Pruning is not able to fill the missing values as accurately as the methodologies related to the SOM.

Computational time is also the smallest when using the Ensemble of SOMs, largely due to the lack of lengthy validation procedure. It can also be seen that EOF methodologies have very high computational time, most likely due to the enormous size of the dataset.

In both presented results in the two datasets, the Ensemble of SOMs is the fastest among the tested methodologies. Similarly, in both presented test cases, the Ensemble of SOMs improve the traditional SOM imputation in terms of accuracy. Furthermore, the computational time is always lower with the Ensemble of SOMs than with the traditional one.

# Chapter 5

# Conclusions

This paper demonstrates many benefits of using the Ensemble of SOMs instead of traditional SOM methodology. The combination achieves better performance than any individual SOM, based on the obtained test errors. Furthermore, the performance of the Ensemble of SOMs is close to or even exceeds the performance of the state-of-the-art methodologies.

In terms of computational time, the Ensemble of SOMs is clearly faster than other compared methodologies in the presented test cases. This is due to the straightforward calibration of the ensemble without the need of lengthy and time taking validation procedure.

Since there are several separated SOM maps to be built, it is also very straightforward to parallelize the computation procedure. This would make the computation even faster without the need to make any changes to the ensembling technique.

At the same time, the EOF Pruning has been proven to be very accurate imputation methodology. Therefore, as a further work, it will be tested the use of the Ensemble of SOMs as an initialization for the EOF Pruning methodology. Since the Ensemble of SOMs is very fast, the computational time of the combination should not increase too much, but the accuracy could be enhanced.

# BIBLIOGRAPHY

[1] SOM Toolbox: http://www.cis.hut.fi/projects/somtoolbox/.

[2] A. Alvera-Azcarate, A. Barth, J. M. Beckers, and R. H. Weisberg. Multivariate reconstruction of missing data in sea surface temperature, chlorophyll and wind satellite fields. *Journal of Geophysical Research*, 112(C03008), 2007.

[3] Marie Cottrell and Patrick Letrémy. Missing values: Processing with the kohonen algorithm. In *Applied Stochastic Models and Data Analysis, Brest, France, 17-20 May*, pages 489–496, 2005.

[4] T. G. Dietterich. *Handbook of brain theory and neural networks*. Cambridge MA: MIT Press, 2nd edition, 2002. Chapter: Articles: Ensemble Learning.

[5] Alexander Ilin and Tapani Raiko. Practical approaches to principal component analysis in the presence of missing values. Technical report, Helsinki University of Technology, Department of Information and Computer Science, 2008. Available from http://lib.tkk.fi/Reports/2008/isbn9789512294824.pdf.

[6] Teuvo Kohonen. *Self-Organizing Maps*. Springer-Verlag, Berlin, 1995.

[7] Teuvo Kohonen. *Data Management by Self-Organizing Maps*, volume 5050/2008, pages 309–332. Springer, Berlin/Heidelberg, 2008.

[8] C. L. Lawson and R. J. Hanson. *Solving least squares problems*. SIAM Classics in Applied Mathematics, 3rd edition, 1995.

[9] Y. Miche, E. Eirola, P. Bas, O. Simula, C. Jutten, A. Lendasse, and M. Verleysen. Ensemble modeling with a constrained linear system of leave-one-out outputs. In Michel Verleysen, editor, *ESANN2010: 18th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 19–24, Bruges, Belgium, April 28–30 2010. d-side Publications.

[10] A. Sorjamaa, A. Lendasse, Y. Cornet, and E. Deleersnijder. An improved methodology for filling missing values in spatiotemporal climate data set. *Computational Geosciences*, 14:55–64, January 2010.

[11] Shouhong Wang. Application of self-organising maps for data mining with incomplete data sets. *Neural Computing and Applications*, 12(1):42–48, 2003.

[12] Z. Zhou, J. Wu, and W. Tang. Ensembling neural networks: many could be better than all. *Artif. Intell.*, 137(1-2):239–263, 2002.

## TKK REPORTS IN INFORMATION AND COMPUTER SCIENCE

| | |
|---|---|
| TKK-ICS-R23 | Matti Pöllä, Timo Honkela, Teuvo Kohonen<br>Bibliography of Self-Organizing Map (SOM) Papers: 2002–2005 Addendum.<br>December 2009. |
| TKK-ICS-R24 | Timo Honkela, Nina Janasik, Krista Lagus, Tiina Lindh-Knuutila, Mika Pantzar, Juha Raitio<br>Modeling communities of experts. December 2009. |
| TKK-ICS-R25 | Jani Lampinen, Sami Liedes, Kari Kähkönen, Janne Kauttio, Keijo Heljanko<br>Interface Specification Methods for Software Components. December 2009. |
| TKK-ICS-R26 | Kari Kähkönen<br>Automated Test Generation for Software Components. December 2009. |
| TKK-ICS-R27 | Antti Ajanki, Mark Billinghurst, Melih Kandemir, Samuel Kaski, Markus Koskela, Mikko Kurimo, Jorma Laaksonen, Kai Puolamäki, Timo Tossavainen<br>Ubiquitous Contextual Information Access with Proactive Retrieval and Augmentation.<br>December 2009. |
| TKK-ICS-R28 | Juho Frits<br>Model Checking Embedded Control Software. March 2010. |
| TKK-ICS-R29 | Miki Sirola, Jaakko Talonen, Jukka Parviainen, Golan Lampi<br>Decision Support with Data-Analysis Methods in a Nuclear Power Plant. March 2010. |
| TKK-ICS-R30 | Teuvo Kohonen<br>Contextually Self-Organized Maps of Chinese Words. April 2010. |
| TKK-ICS-R31 | Jefrey Lijffijt, Panagiotis Papapetrou, Niko Vuokko, Kai Puolamäki<br>The smallest set of constraints that explains the data: a randomization approach. May 2010. |
| TKK-ICS-R32 | Tero Laitinen<br>Extending SAT Solver With Parity Constraints. June 2010. |

ISBN 978-952-60-3247-4 (Print)
ISBN 978-952-60-3248-1 (Online)
ISSN 1797-5034 (Print)
ISSN 1797-5042 (Online)