IMPUTATING MISSING VALUES IN DIARY RECORDS OF SUN-EXPOSURE STUDY

A. Szymkowiak[1], P.A. Philipsen[2], J. Larsen[1],
L.K. Hansen[1], E. Thieden[2], H.C. Wulf[2]

[1]Informatics and Mathematical Modelling, Building 321
Technical University of Denmark, DK-2800 Lyngby, Denmark
Phone: +45 4525 3900,3923,3889
Fax: +45 4587 2599
E-mail: asz,jl,lkh@imm.dtu.dk
Web: eivind.imm.dtu.dk

[2]Department of Dermatology, Bispebjerg Hospital
University of Copenhagen, Bispebjerg Bakke 23
DK-2400 Copenhagen, Denmark

Abstract. In a sun-exposure study, questionnaires concerning sun-habits were collected from 195 subjects. This paper focuses on the general problem of missing data values, which occurs when some, or even all of the questions have not been answered in a questionnaire. Here, only missing values of low concentration are investigated. We consider and compare two different models for imputating missing values: the Gaussian model and the non-parametric K-Nearest Neighbor model.

INTRODUCTION

The missing data problem occurs in virtually any application of statistics to real life problems. It is particularly important whenever statistical analysis is based on human responses. Attempts to fill in missing data ranges from complex monte carlo procedures, like multiple imputation [4], over EM-based, deterministic, yet iterative, procedures [1, 2, 5, 6], to basic statistical methods based on simple multivariate parametric, typically Gaussian, density approximations [3].

In the sun-exposure experiment studied, questionnaires concerning sun-habits were collected from 195 subjects (the group of people involved in a 138 days lasting experiment). In addition, UV radiation were measured at a 10 minute sampling rate. While the ultimate objective is to relate sun-habits, UV dose, and risk of cancer, this work focuses on imputating missing

questionnaire values. We present the analysis of two basic missing values approaches based on parametric and non-parametric representations, respectively. Rather than invoking complex statistical methods, we concentrate on evaluating the two schemes using a modern learning theory tool, the "learning curve", which in the present context quantifies the fill-in error as function of training sample size. Such analysis is important for experimental design. Secondly, we investigate the utility of voting schemes for enhancing the performance of missing data mechanisms.

DESCRIPTION OF THE DIARY DATA

In the experiment two types of data was collected. The subjects wore a special designed watch called the "Sunsaver", which measured UVA and UVB radiation. In addition, the following questionnaire was also returned:

1. Using Sunsaver (yes/no)
2. Working (yes/no)
3. Abroad (yes/no)
4. Sun Bathing (yes/yes-solarium/no)
5. Naked Shoulders (yes/no)
6. On the Beach/On the water (yes/no)
7. Using Sun Screen (yes/no)
8. Sun Factor Number (no/1-7/8-16/17-35/>35)
9. Sunburned (no/red/hurts/blisters)
10. Size of Sunburn Area (no/little/medium/large)

Each questionnaire was stored along with date and subject identification number. Some of the answers are binary (yes/no) whereas others are coded using a 1-out-of-c binary representation. The 1-out-of-c coding ensures that the Hamming distance between any two data vectors equals one, which prevents the introduction of an arbitrary distance for categorical data such as Sunburned.

The sun factor number (question no. 8) has a larger range of values. In order to decrease the length of its binary representation, it is quantized into five levels (no/small (1-7)/medium (8-16)/large (17-35)/huge (>35)). Furthermore, it is combined with question no. 7 creating one binary vector block.

Eventually, for every person and every day, a 17-dimensional binary vector is created. It contains nine blocks from one to four bits each. There are 24212 data records in the diary, distributed among 195 persons and 138 days. There is at least one missing value in more than 1000 vectors due to partially unfilled questionnaires (i.e., in approx. 4% of the questionnaires) which leaves approx. 23000 complete records.

MISSING DATA MODELS

The d-dimensional binary feature vector is defined as $\boldsymbol{x} = [x_1, x_2, \ldots, x_d]$. The data set is denoted as $\mathcal{D} = \{\boldsymbol{x}^{(n)}; n = 1, 2, \ldots, N\}$, where N is the number of questionnaires.

Two models for filling in missing data are described here. The first method is based on the assumption that the diary data vectors are Gaussian distributed. The second is a non-parametric K-Nearest Neighbor model. Many different models can be proposed, however, this paper focuses on comparing a complicated stochastic model with a simpler non-parametric one for specific diary records.

Due to the characteristics of data, there are three different profiles taken into consideration. The first, called the *Complete Diary Profile*, uses the full data set in the estimation. The second *Personal Profile* assumes that questionnaires from one person have similar characteristics while the characteristics across the persons differ. This arise from the expectation that human behaviour varies from person to person. The third profile is the *Day Profile*, which assumes that data vectors for one day are similar or equivalently belonging to one distribution while parameters of the distributions across the days vary. This is due to the fact that human behavior is influenced by weather, temperature, the season of the year, etc. The model using one from the described profiles is called a method.

In addition, a *Voting* procedure is also considered. It compares proposals from all the above mentioned methods and takes the majority vote among the outcomes. This method is expected to give the best results, however, it is much more computationally expensive since it combines the other three methods.

Gaussian Model (GM)

Assume that $\boldsymbol{x}$ is Gaussian distributed with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. Further that the feature vector is divided into observed and missing parts, as $\boldsymbol{x} = [\boldsymbol{x}_o, \boldsymbol{x}_m]$. Under the Gaussian model assumption, the optimal inference of the missing part is given as the expected value of the missing part given the observed part, i.e.,

$$E(\boldsymbol{x}_m|\boldsymbol{x}_o) = \boldsymbol{\mu}_m + \boldsymbol{\Sigma}_{mo}\boldsymbol{\Sigma}_{oo}^{-1} \cdot (\boldsymbol{x}_o - \boldsymbol{\mu}_o) \tag{1}$$

where

$$\boldsymbol{\mu} = [\boldsymbol{\mu}_o, \boldsymbol{\mu}_m] \quad \text{and} \quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{oo} & \boldsymbol{\Sigma}_{om} \\ \boldsymbol{\Sigma}_{om}^{\top} & \boldsymbol{\Sigma}_{mm} \end{bmatrix} \tag{2}$$

The Gaussian imputation model is then given as:

GM Algorithm:

1. Divide the data set $\mathcal{D}$ into two parts. Let the first set contain data vectors in which at least one of the features is missing, call it $\mathcal{D}_m$. Then the remaining part, where all the vectors are complete is called $\mathcal{D}_c$.

2. Estimate mean $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$ from $\mathcal{D}_c$, i.e.,

$$\widehat{\boldsymbol{\mu}} = \frac{1}{N_c} \sum_{n \in \mathcal{D}_c} \boldsymbol{x}^{(n)}, \quad \widehat{\boldsymbol{\Sigma}} = \frac{1}{N_c - 1} \sum_{n \in \mathcal{D}_c} \left(\boldsymbol{x}^{(n)} - \widehat{\boldsymbol{\mu}} \right) \left(\boldsymbol{x}^{(n)} - \widehat{\boldsymbol{\mu}} \right)^{\top} \quad (3)$$

   where $N_c = |\mathcal{D}_c|$ is the number of complete vectors.

3. For each vector $\boldsymbol{x} \in \mathcal{D}_m$

   - Divide the vector into two parts $\boldsymbol{x} = [\boldsymbol{x}_o, \boldsymbol{x}_m]$, where $\boldsymbol{x}_o$ is the observed vector features and $\boldsymbol{x}_m$ the missing vector features.
   - Estimate the missing binary vector as the sign of the conditional-distribution mean for the missing part given the known features:

$$\widehat{\boldsymbol{x}}_m = \text{sign} \left[\widehat{\boldsymbol{\mu}}_m + \widehat{\boldsymbol{\Sigma}}_{mo} \widehat{\boldsymbol{\Sigma}}_{oo}^{-1} \cdot (\boldsymbol{x}_o - \widehat{\boldsymbol{\mu}}_o) \right]$$

K-Nearest Neighbor Model (KNN)

The distance measure for binary vectors (Hamming distance) is defined as follows:

$$D(p, q) = \sum_{i=1}^{d} |x_i^{(p)} - x_i^{(q)}|, \quad (4)$$

where p and q are two binary vectors and i is a bit (dimension) index.

The algorithm for the non-parametric K-Nearest Neighbor Model is given as:

KNN Algorithm:

1. Divide the data set $\mathcal{D}$ into two parts. Let the first set contain data vectors in which at least one of the features is missing, $\mathcal{D}_m$. The remaining part where all the vectors are complete is called $\mathcal{D}_c$.

2. For each vector $\boldsymbol{x} \in \mathcal{D}_m$:

   - Divide the vector into observed and missing parts as $\boldsymbol{x} = [\boldsymbol{x}_o, \boldsymbol{x}_m]$.
   - Calculate the distance Eq. (4) between the $\boldsymbol{x}_o$ and all the vectors from the set $\mathcal{D}_c$. Use only those features in the vectors from the complete set $\mathcal{D}_c$, which are observed in the vector $\boldsymbol{x}$.
   - Use the K closest vectors (K-nearest neighbors) and perform a majority voting estimate of the missing values.

EXPERIMENTS

In order to compare the performance of the models on the diary records, a validation set was taken out from the fully completed questionnaires. We perform a leave-one-out permutation estimate of the generalization error as in 500 repeated permutations one validation sample is chosen randomly from the complete data set, then a number of training samples. The performance is then averaged over the 500 permutations. As an example if considering the Day Profile the day number of the validation sample specifies the day number of the training samples of which there are at most 194 persons to choose from. When training set size, N, is smaller than 194 we randomly choose N out of 194.

We are investigating errors of low concentration, i.e., only one block (question) in the vector is missing at the time. The final error rate is an average over such single errors made in all possible nine blocks.

In the case of the KNN, the model number of nearest neighbors is optimized separately for each profile and for each block using another set of 500 repeated permutation samples. The optimal K in the range 1–30 is then found by picking the one which has the lowest leave-one-out error.

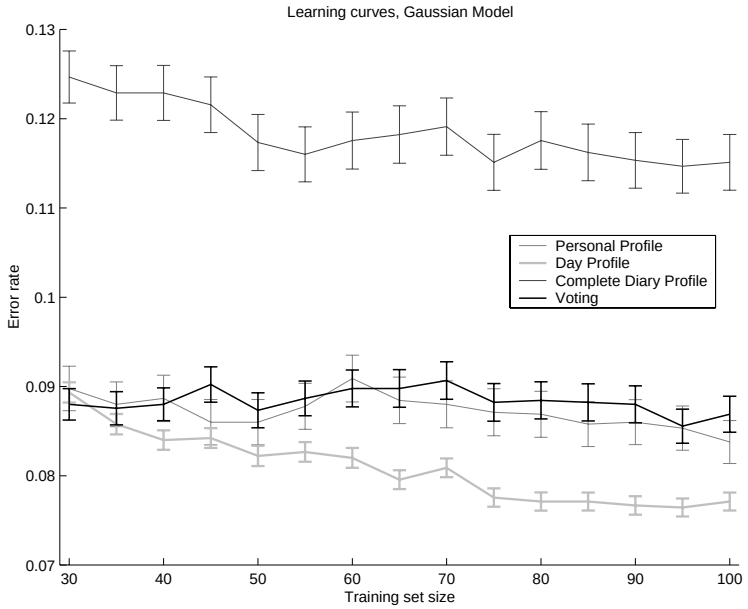

Figure 1: Learning curves for the Gaussian model. Four different methods are presented here: Personal Profile, Day Profile, Complete Diary Profile and Voting. Error bars show deviation from the mean curve over 500 runs.

Figure 1 and figure 2 presents learning curves for the Gaussian model and the K-Nearest Neighbor model, respectively. The deviation from the mean is shown with the error. It decreases slightly with increasing training set size.

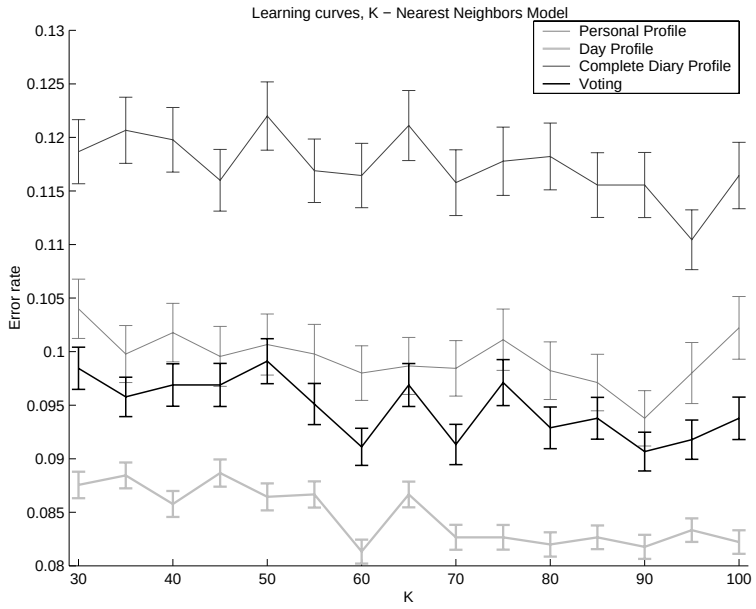

Figure 2: Learning curves calculated for the K-Nearest Neighbor model. Four different methods are presented here: Personal Profile, Day Profile, Complete Diary Profile and Voting. Error bars show deviation from the mean curve over 500 runs.

Voting, as it was expected, gives very good results both for Gaussian model and K-Nearest Neighbor model, however in this case Day Profile performers the best for every training set size.

Figure 3, basically, presents the same as in figures 1 and 2, however with the focus on comparison. Clearly, for large training sets and every profile, the Gaussian model (light) performs better than KNN model (dark).

Figure 4 and figure 5 presents the error rate separately for each of the nine blocks. Every sub-figure corresponds to one question in the questionnaire. For both models the learning curve for block no. 2, which is "Working", (middle-top sub-figure) presents the highest error rate. The error rate for this block basically creates the overall error rate for the validation sample. Not surprisingly, the value of this field is best predicted by Day Profile. For the rest of the blocks, Personal Profile imputate with the smallest error. The situation is similar for the KNN model. However, it is possible to see (also from the figures 1, 2 and 3), that the error rate does not decrease much with increased size of the training set.

Table 1 presents error correlation matrices for Gaussian and K-Nearest Neighbor model, respectively, for three methods. The E_{ij} entry of error correlation matrix is defined as $E_{ij} = \text{Prob}\{\text{error in method } i \wedge \text{error in method } j\}$ estimated by the number of examples where errors occured both in methods i and j relative to the total number of examples. As earlier, each example contains only one block error, and all possible block errors are examined. If only one method out of 3 makes an error it will be corrected by Voting. That

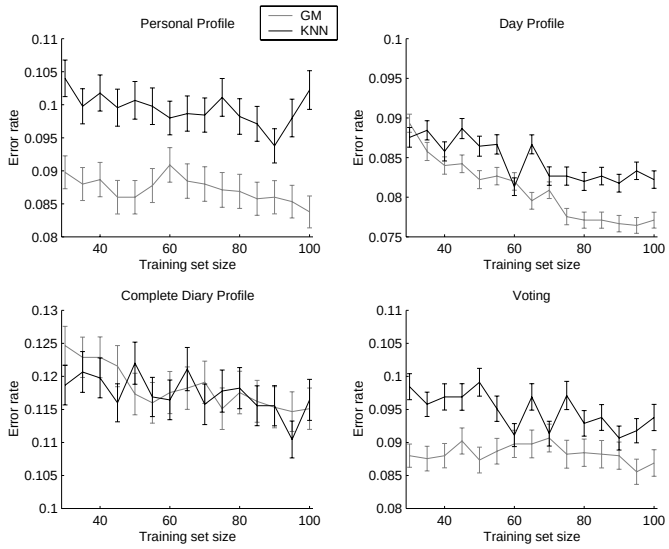

Figure 3: Comparison between GM (light line) and KNN model (dark line) for all the profiles shown separately.

is, the error made by Voting, is given by $\sum_{j>i, i\neq j} E_{ij} + P_3$, where P_3 is the probability of all 3 methods, simultaneously making an error. The gain in error rate by using Voting relative to one of the other methods is given as

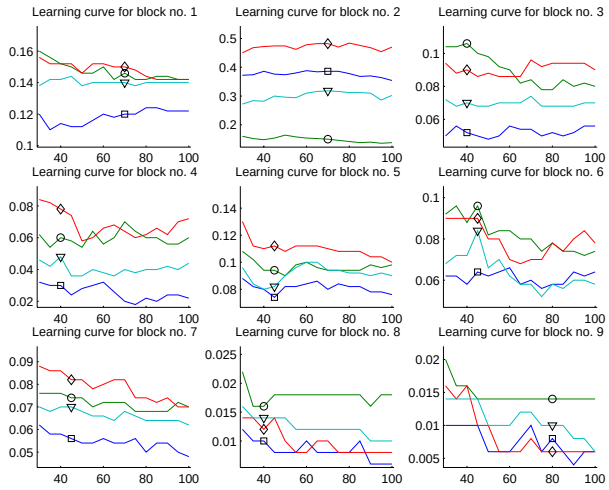

Figure 4: Learning curves for GM model shown separately for all 9 blocks. Block is defined as answer to the question represented binary. On x-axes size of the training set is shown and on the y-axes is error rate. Learning curves for the block no. 2 present the highest error rate. In this case, Day Profile gives the best results in imputating. In the other cases Personal Profile performs the best.

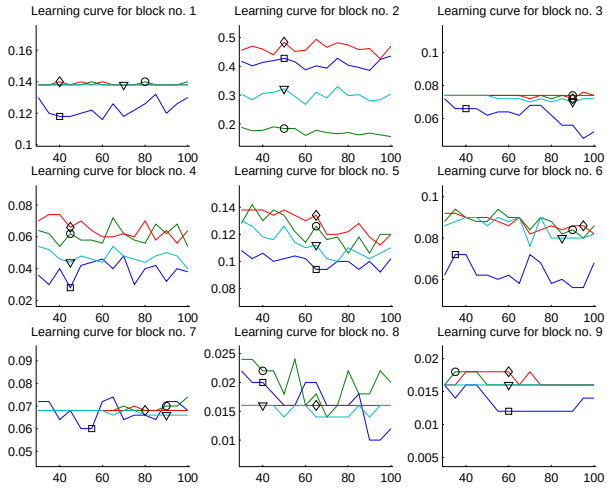

Figure 5: Learning curves for GM model shown separately for all 9 blocks. Block is defined as answer to the question represented binary. On x-axes size of the training set is shown and on the y-axes is error rate. Learning curves for the block no. 2 present the highest error rate. In this case, Day Profile gives the best results in imputating. In the other cases Personal Profile performs the best.

$E_{jk} - E_{ii}$, where $(j,k) \neq i \wedge k > j$.

The error rates are shown for two extreme training set sizes, namely 30 and 100 samples.

Figure 6 shows the error rate for 50 validation samples as a function of training set size. All the methods share the same set of validation samples. Color bars show error rate. It is interesting to see that for some of the validation samples, the error does not depend on which method or model is used, nor the size of the training set (see the sample no. 36). In other cases, increased size of the training set reduces the error rate (sample no. 14

	PP	DP	CDP
PP	0.0271	0.0051	0.0229
DP	0.0051	0.0238	0.0258
CDP	0.0229	0.0258	0.0413
P_3	0.0347		

	PP	DP	CDP
PP	0.0187	0.0040	0.0267
DP	0.0040	0.0169	0.0218
CDP	0.0267	0.0218	0.0322
P_3	0.0344		

	PP	DP	CDP
PP	0.0260	0.0056	0.0227
DP	0.0056	0.0109	0.0222
CDP	0.0227	0.0222	0.0258
P_3	0.0480		

	PP	DP	CDP
PP	0.0280	0.0060	0.0249
DP	0.0060	0.0127	0.0213
CDP	0.0249	0.0213	0.0287
P_3	0.0416		

Table 1: Error correlation tables for GM (top row table) and KNN (bottom row table). Left and right column tables present data for small training set (30 samples) and large training set (100 samples), respectively. Used abbreviations: PP - Personal Profile, DP - Day Profile, CDP - Complete Diary Profile.

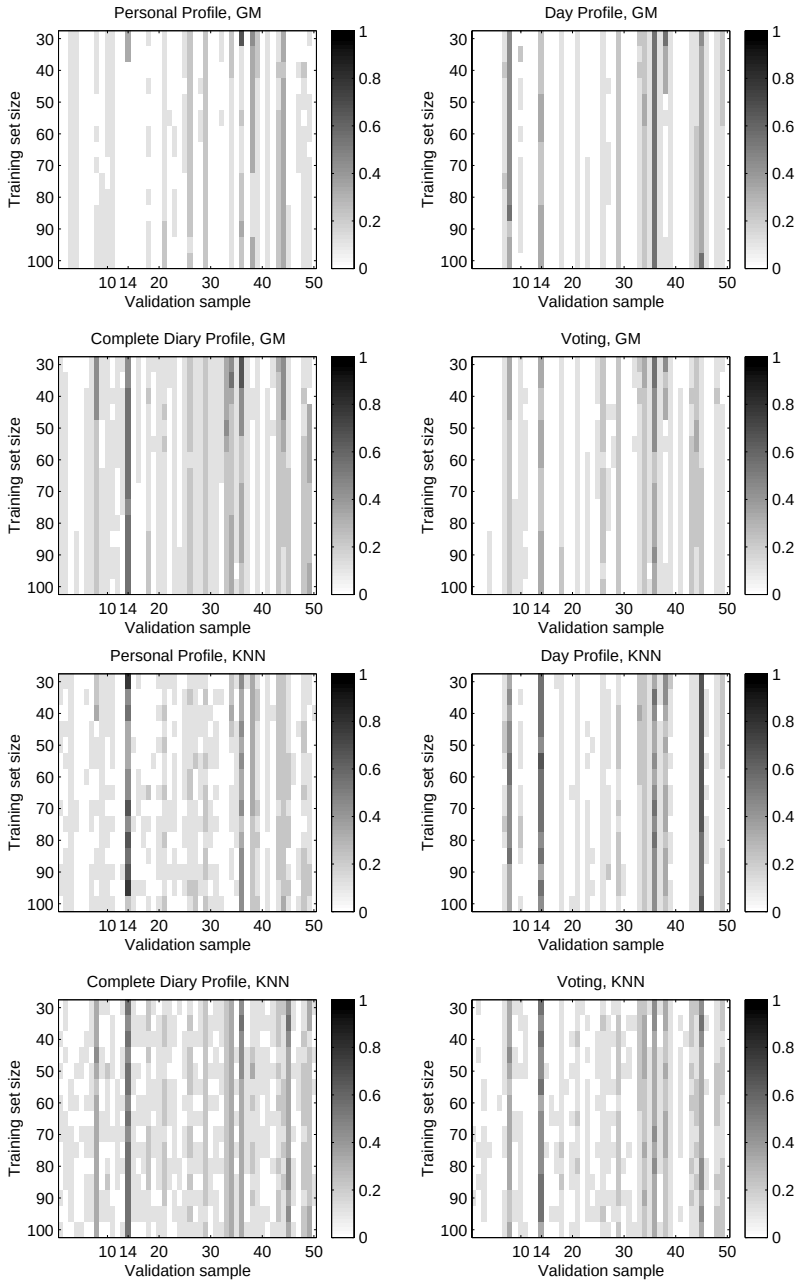

Figure 6: Error rate for different profiles in the GM and in the KNN. Figures show for specific validation samples the dependency on the size of the training set. The methods share the same set of validation samples. Color bars present error rate. 0 corresponds to the situation when for all 9 blocks no error was made and 1 when an error occurred in every block.

for the Gaussian model, Personal Profile). It can also be seen that for other validation samples, the error rate varies from method to method and between the models. In such cases, Voting may return the lowest error rate.

CONCLUSIONS

It is generally expected that the models perform better for large training sets. However, the error rate is strongly sample related, i.e., it can increase significantly with just the one "unlucky" sample.

Applying different methods depending on the block number can be relevant for this data set. In this case using Day Profile in the prediction of the value of block no. 2 and Personal Profile for the rest of the blocks may give considerable improvement in the error rate. However, such mixing of the methods is highly data dependent and has to be tuned manually.

In conclusion, for the present data set, the Gaussian model is superior to the non-parametric K-nearest neighbor model although the Gaussian model assumptions are violated for binary data vectors. The Day Profile method gave best results indicating a strong daily variation. If the errors made by different methods had been uncorrelated, the results returned by the Voting would give the best imputation performance of missing data. For small training sets Voting resulted in improved performance, while severe correlation among the errors of the methods disfavors Voting for large training sets. In addition, the use of overlapping training sets additionally improved correlation among the methods.

REFERENCES

[1] Z. Ghahramani and M. I. Jordan, "Supervised learning from incomplete data via an EM approach," in J. D. Cowan, G. Tesauro and J. Alspector (eds.), **Advances in Neural Information Processing Systems**, Morgan Kaufmann Publishers, Inc., 1994, vol. 6, pp. 120–127.

[2] Z. Ghahramani and M. I. Jordan, "Mixture models for Learning from incomplete data," in T. P. R. Greiner and S. Hanson (eds.), **Computational Learning Theory and Natural Learning Systems**, Cambridge, MA: The MIT Press, 1997, vol. IV: Making Learning Systems Practical, pp. 67–85.

[3] R. Little and D. Rubin, "Statistical Analyses with Missing Data," 1987.

[4] D. Rubin, "Multiple Imputation for Nonresponse in Surveys," 1987.

[5] V. Tresp, S. Ahmad and R. Neuneier, "Training Neural Networks with Deficient Data," in J. D. Cowan, G. Tesauro and J. Alspector (eds.), **Advances in Neural Information Processing Systems**, Morgan Kaufmann Publishers, Inc., 1994, vol. 6, pp. 128–135.

[6] V. Tresp, R. Neuneier and S. Ahmad, "Efficient Methods for Dealing with Missing Data in Supervised Learning," in G. Tesauro, D. Touretzky and T. Leen (eds.), **Advances in Neural Information Processing Systems**, The MIT Press, 1995, vol. 7, pp. 689–696.