Classification of Incomplete Data by Observation

Pierre Lorrentz, *Member, IAENG*

***Abstract* - There are occasions in which databases have feature values that are missing due to errors, irregularities, or unavailable data. Most current imputation methods address cases in which there are sufficient known data to infer an estimate of the missing feature data. This paper proposes a novel imputation algorithm that provides a reasonable solution for the problem domain, represented by databases with missing numerical feature values. This method derives imputed values by observing system configurations and parameters. Given an appropriate model, databases may also be observed. The proposed algorithm employs a weightless multi-classifier that is designed to process certain benchmark databases. Finally, the experiments demonstrate that databases with missing feature values and imbalanced data distributions can still be used effectively.**

***Index Terms* - Incomplete data, Observation algorithm, System configuration, Weightless multi-classifier.**

I. Introduction

An observation can be referred to as the recording of a measurable quantity of a system. A system can be represented in the form of databases. Databases are often discarded due to missing features. It is also commonly claimed that missing data imply missing information. It is possible, however, to a considerable extent, to gain full knowledge of the information that is contained in a given database despite the missing features. This paper explores the issues that govern this problem domain.

The area of data imputation has attracted much attention. Current state-of-the-art methods impute feature values by deductions that are made from existing known data. The state-of-the-art methods work well when sufficient amounts of data are available. For databases in which the features are represented by few or insufficient feature values, state-of-the-art methods or similar methods can lead to biased decisions. The research in this paper explores other alternatives by observing system configurations and parameters to impute missing feature values, which may not be an estimate or zero. It is noteworthy that the methodology that is presented can be applied beyond the case in this report. The case that is presented here involves the implementation of the algorithm in a weightless multi-classifier for testing purposes.

Because classifiers are typically designed to use complete data that are specified at their input, it is difficult for a weightless multi-classifier to perform well when used for databases that contain missing features. The choice of base classifiers for the multi-classifier stems from the need to mitigate imbalances in class distribution, if such imbalances in class distribution are significant and not required. Imbalance class distribution typically occur whenever, using a two-class example, there are many samples of one class, the majority class, and few samples of the other class, the minority class. This phenomenon occurs naturally in, for example, fraud detection, risk management, and medicine. It is often potentially dangerous to ignore minority classes.

Real-life applications produce real data, such as those at http://archive.ics.uci.edu/ml/datasets.html. A real-life application that produces no data might not have a history. There are various reasons why values for a feature might be missing, effecting the grouping of data into categories [1], [2], [3]. The categories are as follows:

Missing Completely At Random (MCAR):- MCAR refers to a situation wherein a missing feature value is not related to variables, methods, or mechanisms that are used to acquire the data. In this case, it is difficult to ascertain when, why, or how the data are missing.

Missing At Random (MAR): This is a situation in which the feature value is missing at random, for which the reasons are traceable, such as process defects and lack of maintenance. Although the source of the missing data is known, as in Shen [4], it might not follow a specific pattern.

Missing Not Completely At Random (MNAR): Missing not at random refers to a situation wherein the source and reason for the missing feature value is known. It is possible to quantify and estimate the present and future amounts of missing feature values, respectively. Because it is possible to state the position and value (estimate) of the missing data precisely, such missing data is often referred to as non-ignorable [1], [5], [6] or informatively missing (IM).

The current state-of-the-art methods that are used to impute the missing feature values or minimise the effects of the missing feature value are divided into value estimation methods and neural networks:

Value Estimation: Value estimation methods model the data generator using the available data and represent the generator of the data as a density function or mixtures of density functions. A good candidate of this method is the Gaussian mixture model. This method requires the availability of large amounts of data from the real generator, or the error in the estimated value for the missing feature value becomes very large. A single value estimate is referred to as a simple imputation. Many and various values for a missing feature value often involve testing each value and estimating the error of a certain objective function. This advanced method includes the multivariate Gaussian mixture model and Expectation-Maximisation (EM) methods. The many-valued estimate methods are referred to as multiple imputations. Similarly, the method in this report is also multi-valued but might not depend on an available known database that is meant for processing, as detailed in section II.

Neural Networks: Conventional neural networks, such as multi-layered Perceptron (MLP) and the Radial Basis Function (RBF) network, can be used singly or in a multi-expert system to acquire information in the databases, despite the missing feature values. At missing data points, a special neural network is used or the activation function is set to

Pierre Lorrentz, Department of Electronics, University of Kent, Canterbury, Kent, CT2 7NT, UK Tel. 0044(0)7813089916.
e-mail: plpress2010@googlemail.com

zero. Several methods of neural networks [7], [8] that also estimate missing feature values exist. Muňoz [9] designed a similarity neurone that ignores feature values.

This report is arranged as follows. Section II describes the observation algorithm, and section III discusses the multi-classifier with which the algorithm is tested. The experiments are described in section IV and use actual databases that correspond to real-life problem domains. The results are presented and analysed in section V. Section VI evaluates the algorithm and compares it with other imputation methods. Section VII concludes.

II. The Observation Algorithm

A. Background

A *measure space* (M, A, μ) is a set, M, with subsets of M. A measure μ on this space is a rule that assigns a non-negative number μ(A) (including ∞) to each subset A in M. If M = ℝ, the real-number line, the definition of *Lebesgue measurable* sets is required on M, e.g., if ρ is any integrable function with∫ ρ(x)dx = 1, then ρ determines a probability measure ρdx on ℝ.

Definition I: Let (M, A, μ) be a measure space*;* a state ρ of (M, A, μ) is a probability measure on M of the form ρμ (ρ ≥ 0), where ρ corresponds to an integrable function such that ∫ρ(x)dx = 1, i.e., ρμ(A) = $\int_A$ ρμ. For any system (M, A, μ) and any state ρ, entropy is defined as:

ρ_{ent} = - $\int_{-\infty}^{\infty} \rho \log \rho \partial \mu$. The *equilibrium state* of the system (M, A, μ) is the position of maximum $\rho_{ent.}$

Definition II: The property or variable of a system that affects the functionality or performance of a system is an observable; therefore, it is worthy of observation. An observable of a system is a numerical, measurable property or variable of the state. If the observable of a system is denoted by y, a map is an ordinary numerically valued function f: M→ℝ, which gives an observable y of space M, a numerical value in ℝ such that for y϶M, f(y)϶ℝ. The observables depend on configurable system parameters.

Definition III: Let y be an observable and ρ be a state; if the observable y is represented as a function, then the expected value of y is defined as $\int_M y.\rho\mu$ with respect to the probability measure ρμ.

* * *

Energy is often defined as force multiplied by distance in the direction of the force. Energy change, dE, is defined as

dE = w * ds

where w represents a "generalised force" and ds is the change in distance in the force direction. With regard to the upcoming discussion, w is replaced by the system under consideration, and ds is also replaced by functions that express the system's parameters. Then, the left hand side dE becomes y, the observed "energy." For an observed energy y (= dE), define the integral

$$f(y) = \int_M e^{-\gamma y} \mu dy \, ; \qquad (1)$$

then, the corresponding state is defined as

$$\rho_y = \frac{1}{f(y)} e^{-\gamma y} \qquad (2)$$

From the state ρ_y, the corresponding entropy can be defined. The function f(y) represents the data imputation system, while ρ_y is the imputed value. Note that f(y) depends on y, and y in turn depends on the system configuration. If y is such that f(y)∞<, then ρ $_y$ is an equilibrium state of the system with respect to y.

For multi-variable and multi-dimensional imputations (i.e., multi-imputation), we define the energy y as the vector valued function **y** = [y_1, y_2, $y_{3\ldots}$ y_k]. Then, **y**: M→ $\mathbb{R}^k$ such that y maps a basis function of M onto the vertices of a simplex, $\mathbb{R}^k$. The observed energy y and the states are correspondingly defined as

$$f(y_i) = 1 + e^{-\gamma_i^1 y_i} + e^{-\gamma_i^2 y_i} + \ldots + e^{-\gamma_i^k y_i} \qquad (3)$$

and

$$\rho_{\gamma_{i0}} = \frac{1}{f(y_i)}; \quad \rho_{y_{i1}} = \frac{e^{-\gamma_i^1 y_i}}{f(y_{i1})}; \text{ etc.} \qquad (4)$$

respectively. Here, $\rho_{\gamma_i^j}$ are states that correspond to possible imputation values. The constants γ_i^j ; i = 0,1,2, ..., are shape constants. Generally, they are of the sequence 3/2; 5/2; ..., etc. The constants γ_i^j are derived from thermodynamics and quantum mechanics of matter, the details of which are beyond the scope of this report.

B. The Algorithm

The process of filling in the missing values is as follows:

1. Locate the position of the missing values when all system parameters are zero—that is, before learning begins.
2. Perform a simple imputation. This is a case when y = y_0 is a constant, or zero if the leading coefficient is not a constant.
3. Perform the normal multiple imputations, and while moving away from extrema (Minimum-Maximum) points, perform a multiple imputation. At this stage, learning/recognition must have started.
4. Repeat step 3 until all missing values have been filled.

The observation algorithm is better described by an example that illustrates the basic functionality of the proposed system. Two variants of the Probabilistic Convergent Network (PCN) were developed locally and named the Enhanced Probabilistic Convergent Network (EPCN) [10]

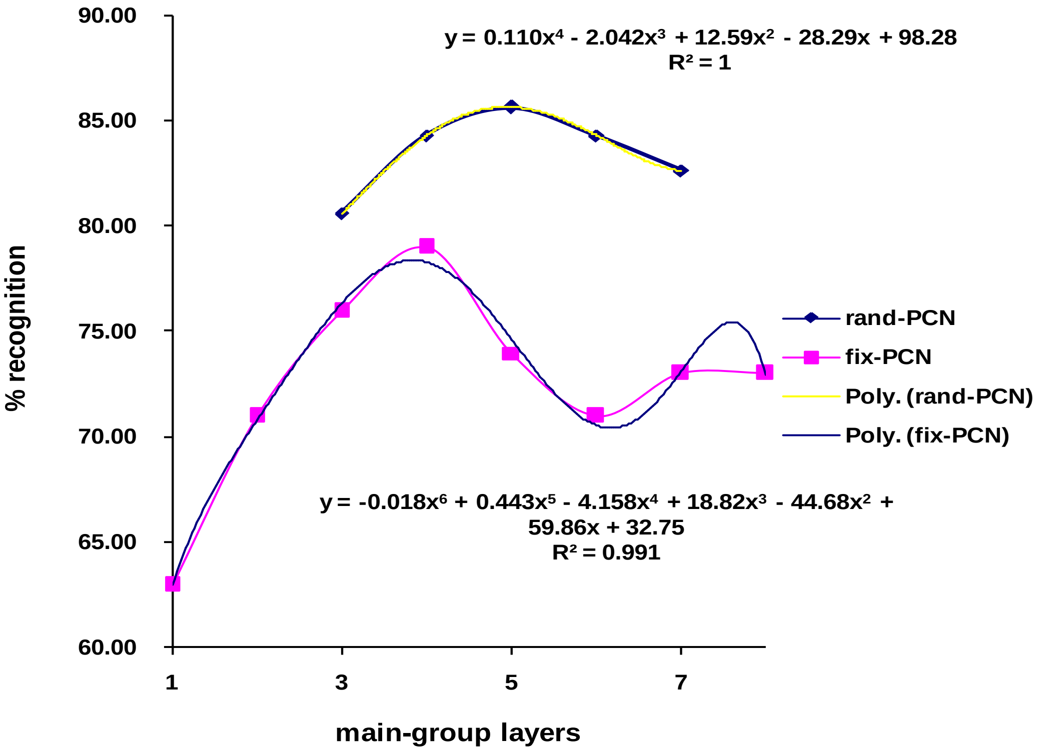

Fig. 1: A plot of the dependence of performance on main-group layers.

When tested on unconstrained handwritten numbers from The Centre of Excellence for Document Analysis and Recognition (CEDAR) [11] and other databases, the performance varied as the number of main-group layers varies. The graph that was obtained from the CEDAR (centre of Excellence for Document Analysis and Recognition) database, as performance varied with the number of main-group layers, is displayed in Fig. 1. The database from CEDAR has no missing feature values, but the variation in main-group layers with respect to performance illustrates imputation values can be deducted from the system configuration. From Fig. 1, it follows that

(a) when $y=98.28$ and $\gamma=(3/2)$

$$\int_{-30}^{30} e^{((-3)/2)98.28}dx = 5.\ 6814\times10^{-63}$$

$g(x)=f(x)^{-1}e^\wedge\{-\gamma y\}$
$f(x)^{-1}e^\wedge\{\gamma y\}=(\ 5.\ 6814\times10^{-63})^{-1}\times e^\wedge\{((-3)/2)98.28\}$
$(\ 5.\ 6814\times10^{-63})^{-1}\times e^\wedge\{((-3)/2)98.28\}=\ 1.\ 6667\times10^{-2}$
$\rho=1.\ 6667\times10^{-2}$

(b) when $y = 98.28-28.29x$ and $\gamma = (3/2)$

$$f(x) = \int_{-30}^{30} e^{((-3y)/2)}dx$$

$g(x) = f(x)^{-1}e^\wedge\{((-3y)/2)\}$
$g(x) = 5.\ 9269\times10^{-488}\exp(42.\ 435x-147.\ 42)$
$[5.\ 9269\times10^{-488}\exp(42.\ 435x-147.\ 42)\ ]\_\{x=29.88\}= 0.26074;\quad \rho= 0.26074$

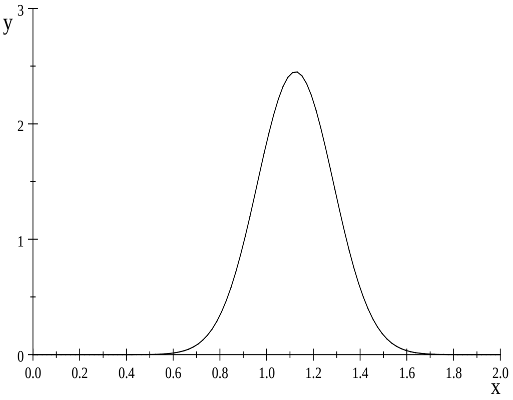

Fig. 2: A plot of y (see Fig. 1) for fix-PCN for power of x=3.

(c) when $y=98.28-28.29x+12.59x^2$ and $\gamma=(3/2)$

$$f(x) = \int_{-30}^{30} e^{((-3y)/2)}dx = 8.\ 7011\times10^{-55}$$

$g(x) = 1.\ 1493\times10^{54}\exp(-18.\ 885x^2+42.\ 435x-147.\ 42) =$
$1.\quad 1493\times10^{54}\exp(-18.\ 885x^2+42.\ 435x-147.\ 42)$
$[1.\ 1493\times10^{54}\exp(-18.\ 885x^2+42.\ 435x-147.\ 42)\ ]$
Candidate(s) for extrema: {2. 4518}, at {[x=1. 1235]}
$[1.\ 1493\times10^{54}\exp(-18.\ 885x^2+42.\ 435x-147.\ 42)\ ]\_\{x=0.8\}= 0.33971\quad \rho=0.33971$

(d) when $y = 98.28-28.29x+12.59x^2-2.042x^3$ and $\gamma = (3/2)$

$$f(x)= \int_{-30}^{30} \exp\left(3.063x^3-18.\ 885x^2+42.\ 435x-147.\ 42\right)dx = 1.\ 2607\times10^{29020}$$

$g(x) = 7.9324\times10^{-29021}\exp(3.063x^3-18.885x^2+42.435x-147.42)$

$7.9324\times10^{-29021}\exp(3.063x^3-18.885x^2+42.435x-147.42)$

$\rho = 0.30983$

(e) when $y=98.28-28.29x+12.59x^2-2.042x^3+0.11x^4$ and $\gamma=(3/2)$

$$f(x) = \int_{-30}^{30} e^{((-3y)/2)}dx$$

$f(x) = \int_{-30}^{30}\exp(-0.165x^4+3.063x^3-18.885x^2+42.435x-147.42)dx = 3.0167\times10^{-51}$

$g(x) = 3.3149\times10^{50}\exp(-0.165x^4+3.063x^3-18.885x^2+42.435x-147.42)$

$3.3149\times10^{50}\exp(-0.165x^4+3.063x^3-18.885x^2+42.435x-147.42)$ $3.3149\times10^{50}\exp(-0.165x^4+3.063x^3-18.885x^2+42.435x-147.42)$

Candidate(s) for extrema: $\{1.1626\times10^{-5}, 9.0649\times10^{-3}, 1.2724\}$,

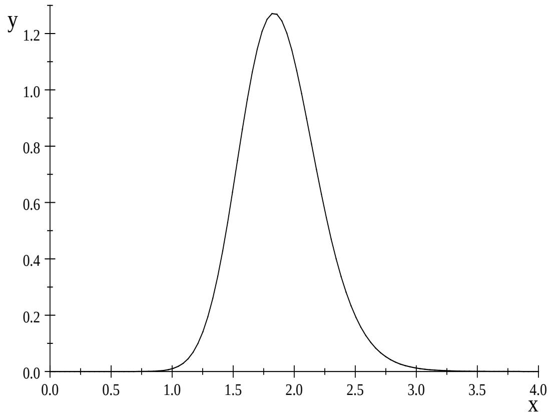

Fig. 3: A plot of y (see Fig. 1) for fix-PCN for power of x=4.

at $\{[x=4.8285],[x=7.2601],[x=1.8341]\}$

$\rho = 6.9548\times10^{-3}$

* * *

An appropriate imputation value minimises performance errors. It may not be a value that estimates the missing feature value. An estimate of the missing feature value would be an appropriate imputation value if y in equations (1) and (2) is replaced by a model of the data generator. Considering the *source* of imputation values, imputation methods can be grouped into three categories. The first category of imputation methods uses available known data to estimate the missing values. This category often uses statistical methods. A notable example is the EM algorithm. The EM algorithm, explained by Hui [7], is interesting. The concept of adaptive imputation is espoused in Hui [7], whereby imputation is performed analytically in the E-step of the expectation (E)-maximisation (M) algorithm of a quadratically gated mixture of experts. Williams [12] integrates out the incomplete data by using an estimated conditional density function.

The second category uses neural networks to account for the missing feature data. The neural network simply isolates the missing feature values from being processed or assigns a safe value to replace missing feature values. Muller [13] uses neuro-fuzzy coding in a multi-classifier design, in which the lack of a value is explicitly coded into the neurons by setting the corresponding activation level to zero. Morris [14] designed a hybrid multi-classifier in which a classifier is dedicated only to missing feature values. The dedicated classifier calculates the expected value of the missing feature values.

The imputation source of the third type does not use the available data or the system that processes the data directly to account for missing feature values. Rather, an arbitrary safe value is simply imputed. Examples include most simple imputation methods. Mohammed [15] uses a "sentinel" value as a simple imputation method in multi-classifier designs.

The grouping according to sources of generating imputation values is not mutually exclusive. The observation algorithm may show that this is the case. The observation algorithm reveals that when y is zero or a constant, the imputation is a simple imputation (example (a)). The situation in which y represents the system that processes the data is shown in examples (a) to (e). The observation algorithm can impute feature values from known data directly if the available data are statistically sufficient to infer a generator for the data. This often occurs in terms of probability distribution function (pdf). Examples of probability distribution functions are normal probability distribution function and Poisson probability distribution function. The normal probability distribution function can generate a RBF. The normal probability distribution function, when Bayes theory is applied, can yield the EM algorithm, which might in turn be used in multi-classifier designs. When a Gaussian distribution is used as y in the observation algorithm, it is able to estimate missing feature values directly from known available feature data with a certain minimum error (see section VI).

C. Notable features of the Observation Algorithm

1) Systems and devices can be modelled by one of the *autoregressive methods* to yield y (in equation (1) and (2)). It is also possible that f(y) = y in equations (1) and (2). The examples and the experiments in this paper should be regarded as specific illustrations of the proposed algorithm.
2) Any error feedback is a property of the system that is being modelled and not directly that of the proposed algorithm; i.e., the algorithm is independent of error feedback.
3) Some imputation methods rely on available data to estimate imputation values, while others insert arbitrary (safe) values. The proposed method is principled and uses available data or the system (model) that processes the data to impute values.

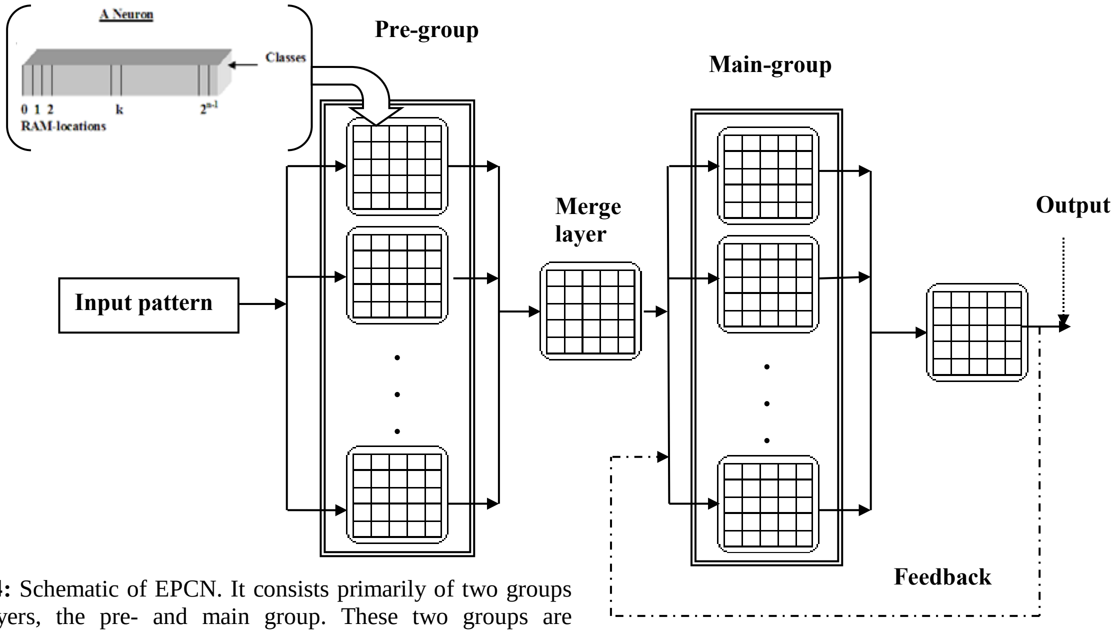

Fig. 4: Schematic of EPCN. It consists primarily of two groups of layers, the pre- and main group. These two groups are separated by a merge layer.

4) The possibility for imputation grows with increasing problem complexity. For 1-dimensional data, as in example (a) (there, y = 98.28), the algorithm behaves like a simple imputation. Most systems, however, depend on more than one independent variable; thus, the algorithm nearly always works in the multi-dimensional domain, which implies an imputation, as demonstrated in example (b) to (e). Expectation-Maximisation (EM) algorithms operate only on databases; they are statistical, and a database needs to be large enough to estimate a missing value accurately. Given that a large database has been modelled (y in equation (1) and (2)) in terms of probability distribution function or mixtures of probability distribution functions, the observation algorithm might behave similarly to EM algorithms and estimate missing values. It is noteworthy that the observation algorithm is not limited to databases or explicit statistical phenomena.

III. MULTI-CLASSIFIER SYSTEMS

A weightless multi-classifier uses weightless neural networks as component classifiers. The combination (fusion) of the component classifiers is achieved by a trained combiner. Component classifiers are derived from an Enhanced Probabilistic Convergent Network (EPCN). EPCNs are n-tuple classifiers whose architecture consists of a pre-group layer, a merge layer for the pre-group, a main-group layer, and a merge layer for the main group, as shown in Fig. 4. Each layer consists of neurons, and each neuron (see Fig. 4) consists of RAM-locations. The learning process of an EPCN consists of deriving n digits from input data and forming d = 2^n addresses.

Depending on the actual values of d, the corresponding RAM location of the pre-group layers is addressed. The recognition process also entails deriving d addresses from the input space and modifying the corresponding RAM memory in the layers of the main group. Learning is supervised, and the recognition results are summed independently per class to output.

Two types of EPCNs are used in this work: fix-EPCN and rand-EPCN [10]. The EPCN is adaptive, and the system parameters of each component EPCN are randomly determined to increase the diversity among them. The weightless multi-classifier is made up of EPCNs in parallel (the [P_i, M_i] pair), as shown in Fig 5, and a probabilistic classifier a further EPCN (the [P_c, M_c] pair) with unique system parameters for the fusion of the component networks. In Figure 5, the gating function, *f (.)*, is a function that not only produces a weighted sum of the component classifier but also encodes this sum to a form that is understandable to the EPCN combiner.

A. Imbalance data distribution

Databases with some missing features can also be imbalanced with respect to class distribution. A phase within the algorithms modifies the frequency of occurrence, in RAM locations, of each majority class to avoid undermining the presence of the minority classes. For N training patterns and x divisions, a frequency of occurrence value “a” that occurs in a memory location will be adjusted as:

$$\hat{a} = a(\frac{x}{N});$$

$$\tilde{a} = round(\hat{a});$$

T

where ã=new vaue replacing a in that location

his step in the learning and recognition algorithm reduces the large probability of the majority classes to accommodate the minority classes. This process also removes rounding errors and truncation errors. A different approach is employed in Yen [8], wherein under-sampling of data in a clustering procedure mitigates the problem of imbalances in data distribution

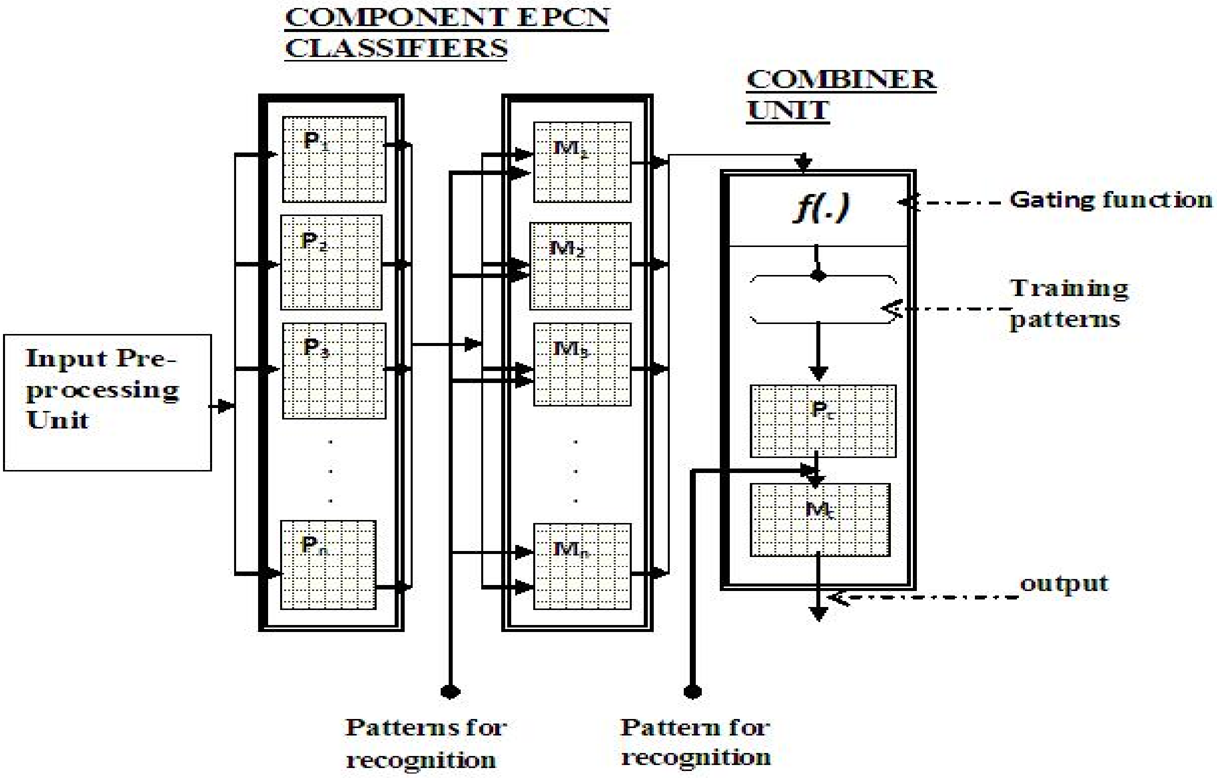

Fig. 5: This is a simplified diagram of the RAM-based multi-classifier. It consists of EPCNs as component classifier.

IV. EXPERIMENTS

The experimentation in this section is designed to explore the behaviour of the observation algorithm when it is exposed to real-life data. All databases that have been completed by the observation algorithm are either the MCAR or MAR type. The first step in the exploration requires artificially and randomly deleting some data from the database, consequently omitting some feature values from the ionosphere database, in subsection IV (A). The deletion of feature values is done randomly well before separation into train and test sets, ensuring that both sets are affected, irrespective of database sub-setting. The observation algorithm and simple imputation were applied to the ionosphere database. The other databases (i.e., in subsections IV(B) and IV(C)) contain naturally occurring missing feature values, and only the observation algorithm is used. Ten benchmark databases are further processed, similar to sections IV(B) and IV(C), with naturally occurring missing feature values, using the observation algorithm for data imputation.

The train set is randomly sampled without replacement from the databases. The bare multi-classifier (see section III) is designed to treat only complete databases. But, when databases with missing features are presented to the multi-classifier, the observation algorithm calculates imputation values from the multi-classifier. The calculation of values from the multi-classifier by the observation algorithm (section II) is independent of the size of the databases but depends on the variable and parameters of the multi-classifier. The observation algorithm is designed to give meaningful responses to databases with a small percentage of missing features.

A. Ionosphere

The source of databases for this experiment was: Space Physics Group, Applied Physics Laboratory, Johns Hopkins University, Johns Hopkins Road, Laurel, MD 20723, U.S.A. [16], [17] . The database has no missing feature values. All missing feature values are created during the experimental exploration. The experimental procedure is as follows:

- Some missing feature value is artificially created in the database.
- The database is divided into two sets: the training set and test set. Training set data are randomly sampled from the database without replacements; thus, the training and test sets form a disjoint set. This partitioning strategy is also shown in Table 1.
- A multi-classifier is initialised, consisting of two EPCNs in parallel, as the component classifier. The component classifiers are fused by a trained combiner, which is an EPCN with unique system parameters.
- The component classifiers are all trained and tested by both the train and test sets shown in Table 1.

Table 1: Partitioning of ionosphere database into train set and test set.

Class	Training set	Test set	Total
1 (b)	38	88	126
2 (g)	68	157	225

- This is a two-class classification task that involves the good (g in column 1, row 2 of Table 1) and bad (b in column 1, row 3 of Table 1) characteristics.
- The amount by which missing values are created is expressed as the percentage (%) of the total amount of feature values. First, 0.4% of the total amount of feature values is deleted, and steps 1–5 above are carried out. Subsequent re-run of steps 1–5 above is performed on the database in which the missing feature value has increased.
- All steps above are repeated for the normal state-of-the-art simple imputation.

B. Post-Operative

Certain places within a medical centre are designated as I, S, and A, where I = intensive care unit; S = go home, and A = general hospital floor (outpatient room). These designated areas are where the patients should go after their operation if they are advised to do so. The classification task of the database that is used is to specify where patients in a recovery area should be advised to go after an operation. The source of the database is: Sharon Summers, Source of Post-operative Database: School of Nursing, University of Nursing, University of Kansas Medical Centre, Kansas City, KS 66160 U.S.A. [18], [19]. This database has less than 0.5% of attributes missing. It also demonstrates the presence of a minority class. The experimental procedures are as follows:

- The database is partitioned into a training set and test set, as summarised in Table 2. The training set is randomly selected from the total data.

Table 2: Partitioning of Post-operative database into train-set, and test-set

Class	Train	Test	Total
1 (I)	1	1	2
2 (S)	8	16	24
3 (A)	20	44	64

- A multi-classifier is initialised and consists of two EPCNs in parallel and a trained combiner for fusion. The component classifiers of the multi-classifiers are all trained and tested by the datasets of the training and test sets, as shown in Table 2.

C. Lung Cancer

The lung cancer data are found at: http://archive.ics.uci.edu/ml/datasets.html [20].

- The database is divided into a training set and test set. The training set is made up of a random selection of the database without replacements. Table 3 summarises the distribution of data among the train and test sets.

Table 3: Partitioning of the lung cancer database into train set and test set

Class	Train	Test	Total
1	3	6	9
2	4	9	13
3	3	7	10

- A multi-classifier is initialised, consisting of two EPCNs as base classifiers. Each EPCN is independently and dynamically configured with a randomisation strategy that is aimed at increasing diversity. The fusion of the component classifier is also accomplished by a trained combiner.
- Both the train and test sets are used equally on each component classifier of the multi-classifier during learning and subsequent testing.

Experiments that are similar to IV(B) and IV(C) are also performed on other benchmark databases, most with naturally occurring feature values (either MCAR or MAR type) from the UCI [20] repository.

V. Evaluation of the proposed System on a Mult-Classifier

The results in this section are summaries of results from the experiments of section IV.

A. The ionosphere database

Fig. 6 shows a plot of the performance for a state-of-the-art simple imputation method and an observation algorithm on the ionosphere database. In addition, the results demonstrate that the observation algorithm gives a reasonable result from databases with missing feature values. However, performance declines as the number of missing feature values increases. The ionosphere database consists of 34 features per instance and a total of 11,934 feature values. As an illustration, 0.5% of ionosphere database is approximately 60 feature values. The rapid decline in performance is due to the use of very few component classifiers (in the multi-classifier). This decline demonstrates the relationship between missing feature values and the tested observation algorithm clearly. If more than 15% of the feature values are missing in the data and a high recognition rate is obtained, such results might be biased. Because there are few component classifiers (2 bc in Fig. 6), each of them sees imputed feature values in place of missing feature values more often, as opposed to when five-base (5 bc in Fig 6) classifiers are initialised. Performance then decreases rapidly, in Fig. 6, when the number of the missing feature values increases.

The aims and objectives of the exploration are not only to achieve a high-level of accuracy but also to demonstrate and compare the effects of the observation algorithm if the number of missing feature values increases. Using the same ionosphere data, the features and their numerical values, both in [15], were systematically removed to investigate the effect of missing feature values on performance.

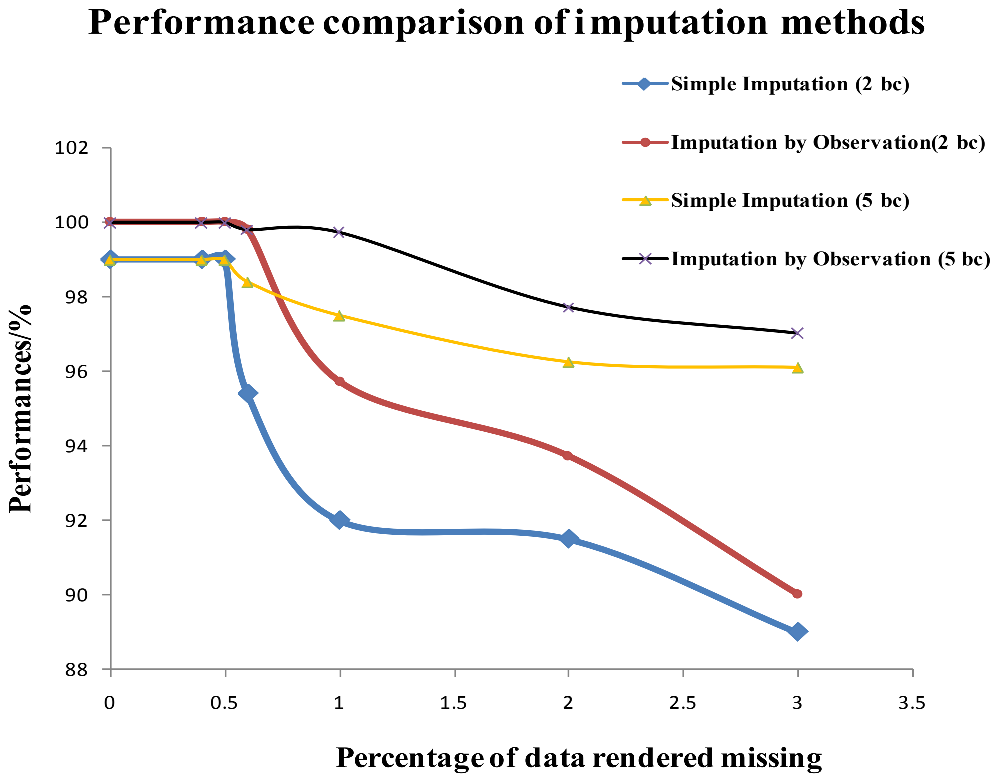

Fig. 6: The percentage of data correctly classified (vertical axis) when attribute values (horizontal axis) are systematically deleted from the ionosphere database. "bc" refers to the base classifiers used. (2 bc) = two-base classifiers and (5 bc) = five-base classifiers.

In that report (i.e., in Mohammed [15]), a performance of 90.7% was recorded at 2.5% missing feature values. This result corresponds loosely to the case in which all features are maintained and a performance of approximately 92% was recorded, in Figure 6, at 2.5% missing feature values. The same rate was obtained by Sigillito [17], using a “non-linear” perceptron with no missing feature values. Fig. 6 also shows that at up to 3.0% of missing feature values, reasonable information in the database is obtainable.

B. Post-Operative

The confusion matrix in Table 4 is obtained when the observation algorithm is tested on post-operative databases. The results, as shown in Table 4, suggest that the proposed algorithm compensates for the effect of missing feature values. With regard to determining a minority class, if a pattern class has less than 10% of data (or patterns) compared with other classes, then that class is in the minority. For example, class I in Table 4 is 8.33% of class S and 3.13% of class A and, thus, is in the minority. The minority class consists of only two instances.

Table 4: Confusion matrix from the RAM-based multi-classifier when the Post-operative database is used. The columns represent the classes (I,S,A) while the row represent instances.

	I	S	A	Unclassified
I	1	0	0	0
S	0	16	0	0
A	0	0	44	0

The percentage of missing feature values in the database was 0.42%, which is less than 0.6%. The type of results in Table 4 is not unexpected. The results also demonstrate the ability of the component classifiers to compensate for the effect of imbalances in data distribution. The post-operative database is characterised by both imbalances in data distribution and missing features; a single classifier yields very low performance e.g., Woolery [21] and Budihardjo [22] recorded a performance rate of approximately 48%.

C. The Lung-Cancer database

The lung cancer database has 5 missing feature values. Missing features are made up of approximately 0.28% of the total attribute values. When the algorithm is tested on the lung cancer database, the confusion matrix in Table 5 is obtained.

A performance of 90.91% (Table 5) suggests that the database is useful and should not be discarded. Although the database is small in size, the detection of pathological lung cancer in a database of this type requires high-level classification accuracy. One reason for this requirement is that this area involves saving human lives. For example, Hong, applying the K-nearest neighbour (K-NN), achieved a 77% accuracy rate [23]. Use of the optimal discriminant plane effects a 59.4% accuracy rate [23].

Table 5: Confusion matrix from the RAM-based multi-classifier when the lung cancer database is used. The column represents the classes (named 1, 2, and 3) and the rows are instances. The last column represents unclassified patterns.

	1	2	3	Unclassified
1	5	0	1	0
2	0	8	0	1
3	0	0	7	0

D. *Other Databases*

The results of 10 benchmark databases with naturally occurring missing feature values from the UCI repository [20] are summarised in Table 6. Imputation values for the missing feature values are derived from the multi-classifier; thus, the size of the databases did not affect the results. But, the number of base classifiers (2, in this case) did affect the results, as explained in subsection V(a). Databases without comments in the comment column had less than (but greater than zero) 5.6% of feature values missing.

Table 6: Summary of results from 10 benchmark databases when the observation algorithm is used (% = percentage).

DATABASES	**RECOGNITION RATE/%**	**COMMENTS**
Autos-import	98	-
Autos-mpg	97.6	-
Breast-Cancer	99.2	-
Bridges	92	5.6% missing
Echocardiogram	94.7	-
Heart-diseases	99.4	-
Hepatitis	95.3	-
Horse colic	87.6	30% missing
Mushroom	98.1	-
Solar-flare	99.4	none missing

VI. Evaluation of the proposed System on Data

Evaluation of the observation algorithm is better achieved by performance, because it might form a component of a system and might not be the speed-determining step of the system. Analytical evaluation of the observation algorithm might become overtly mathematical and might fail to demonstrate its usefulness in any specific application. Thus, in this paper, evaluation of the observation algorithm's performance by experimentation is the optimal choice.

The observation algorithm can derive imputation values from the system or from data. The derivation of imputation values from the system that processes the data is the subject of section II. In this section, the observation algorithm will be evaluated, based on the derivation of imputation values from existing known data. If, in (1)

$\gamma = \frac{1}{\tau_i}$; i = 0,1,2, ..., ∞; then

$$f(y) = \int_M e^{\frac{-y}{\tau_i}} \mu dy \qquad (5)$$

and the corresponding state is defined as

$$\rho_y = \frac{1}{f(y)} e^{\frac{-y}{\tau_i}} \qquad (6)$$

State (6) can be referred to as Boltzmann's distribution function. τ_i is referred to as the temperature parameter or annealing schedule. When the observation algorithm is modified in this manner, it is able to sample from available data directly. But, to achieve this sampling, (6) must be modified slightly to yield the conditional probability with which the data are sampled. Following a method that is similar to Gibb's sampling algorithm [24], (6) becomes

$$\rho_{\tau_i}(y_i \mid n_i) = \frac{e^{\frac{-y_i}{\tau_i}}}{\int_M e^{\frac{-y_i}{\tau_i}} \mu \partial y} \qquad (7)$$

where $n_i = y$, wherein y_i is omitted. All other variables have their usual definitions. Because imputation values should not bias the data that are processed, the following constraint is essential:

- The available known data should be statistically sufficient.

Sampling from previously sampled data is typical during multiple imputations. The imputed value is derived by multiplying a constant or the previous value by the conditional probability (7). Those data that have been sampled by multi-imputation are processed by the multi-classifier. The example in section II illustrates this model when y is set to a Gaussian distribution function. In general, any distribution is adequate if the distribution is an adequate model of the data that are processed.

The experiment uses waveform data from the UCI repository [20]. From waveform data, features are removed prior to separation into the train and test sets. Experiments that are similar to those with the ionosphere database (section IV) are performed on the waveform data to compare the observation algorithm and similar algorithms— specifically, the infinite imputation.

In infinite imputation [25], the distribution that governs the available data is a free parameter, like the observation algorithm. The observation algorithm and infinite imputation use the same database and the same model and treat data models as a free parameter. Uwe [25] uses the waveform database, a Gaussian distribution with no restriction as a model of data, and a Gaussian mixture model (GMM) with 10 centres. Similarly, the observation algorithm uses a Gaussian distribution as a model of waveform data, without any restrictions.

A. *Comparison of performance of imputation methods*

Five-base classifiers constitute the multi-classifier that is used and generates a graph that is similar to Fig. 6, obtained from the waveform [20] database. Research on imputation methods is an emerging field, as evidenced by the lack of a ground truth. For this reason, performance is compared with Weighted Infinite Imputation (WII) in Table 7. The observation algorithm is similar to WII when it processes data for imputation values, because both methods treat data models as a free parameter. The observation algorithm and WII both achieve 100% accuracy for very low missing features. For missing feature values greater than 3%, the observation algorithm outperforms WII, as shown in Table 7.

Table 7: Performance of the observation algorithm and WII imputation methods using Gaussian distribution as the data model.

Imputation methods	% missing	Data model	% perfomance
Observation algorithm	30	Gaussian pdf	91.5
WII [25]	30	Gaussian pdf	86.12

Comparisons are also made between the observation algorithm and other less similar imputation methods in Table 8, based on the ionosphere database. A comparison of the observation algorithm with the state-of-the-art EM method is presented in the table below. The imputation method in [15] is a simple imputation that uses a certain "sentinel" value. In [15], multi-layered perceptron (MLP) was used as a component classifier in a multi-classifier framework and experimented on a benchmark ionosphere database. The results are compared in Table 8.

The MLP-based multi-classifier consists of an ensemble of approximately 1000 classifiers [15], and the observation algorithm uses a weightless multi-classifier that consists of two classifiers as component classifiers on the ionosphere data. The EM algorithm in [26] makes a multiple imputation for missing features. Also, in [26], variants of the EM algorithm for imputation are compared.

VII. Conclusion

An observation algorithm has been introduced, which, when used with a multi-classifier, generates high classification accuracy of incomplete databases. The experimental results that demonstrate high-level accuracy in processing missing feature values indicate that the proposed imputation method is suitable for situations for which multiple imputations are required. The presented observation algorithm is thus suitable for databases with missing features, provided the number of missing feature values is low. Generally, the accuracy depreciates rapidly as the number of missing feature values increases. The results show that the rapid decline in performance as the number of missing feature values increases can be mitigated by increasing the number of component classifiers as demonstrated.

The proposed multi-classifier also demonstrates a solution within its learning/recognition algorithm to the problem of imbalances in data distribution

Table 8: The observation algorithm compared with EM and other imputation methods

Methods	% Missing	Performance/%
Simple Imputation [15]	2.5	90.7
Observation Algorithm	2.5	92.5
Expectation–Maximisation algorithm [26]	25.0	85

REFERENCES

[1] Little R.J., and Rubin D.B., Statistical Analysis with Missing Data, Wiley, New York, 1987.

[2] Burk S.F., A method of estimation of missing values in multi-variable data suitable for use with an electronic computer, J.R Stat. Soc. 1960, B22, 302 - 206.

[3] Fulufhelo V.M, Shakir M. and Tshilidzi M., Missing Data: A Comparison of neural network and expectation maximisation technique, Current Science, Vol. 93, No. 11, Dec. 2007.

[4] Chien-wen Shen, Cycle-time Forecasting Models for Defect Inspection Process in TFT-LCD Module Assembly, Engineering Letters, 16:3, August 2008.

[5] Shafer J.L., Analysis of Incomplete Multivatiate Data, Chapman 7 Hall, new York, 1997

[6] Schafer J.L and Graham J.W.; Missing Data: Overview of the state-of-the-art, Psychol Methods, 2002,7,147 -177.

[7] Xuejun L, Hui L, Lawrence C (2007) Quadratically Gated Mixture of Expert for Incomplete Classification, Int. Conf. On Machine Learning, Corvallis, OR

[8] Yen S-J, Lee Y-S, (2003): Cluster-based under-sampling approaches for imbalanced data distributions, Expert Systems with Applications (2008).

[9] Lluis A.B. muňoz, Ramos J.J.V, Similiarity-based Heterogeneous Neural Networks, Engineering Letters, 14:2 May 2007.

[10] Lorrentz P, Howells WGH, McDonald-Maier KD (2006) EPCN, RASC, pp. 267 – 272

[11] The CEDAR, Handwritten Numerals, University at Buffalo, State University of New York, Department of Computer Science, 1978

[12] David W, Chunping W, Xuejun L, Lawrence C (2007) Classification of Unexploded Ordnance Using Incomplete Multisensor Multiresolution Data, IEEE Transactions on Geoscience and Remote Sensing, Vol. 45, No. 7

[13] Stephanie M, Patrick G, Jean-Denis M, Rene C, and Yves C (1998) A Neuro-fuzzy Coding for Processing Incomplete Data: Application to the Classification of Seismic Events, Neural Processing Letters 8: 83 – 91, Kluwer Academic Publishers.

[14] Andrew C Morris (2000) A Neural Network for Classification with incomplete Data, IDIAP 00-23

[15] Hussein M, Robi P (2006) A Random Subspace for Missing Data, RASC pp. 206 - 216.

[16] Space Physics Group (1989), Database Source, Applied Physics Laboratory, Johns Hopkins University Johns Hopkins Road Laurel, MD 20723, USA

[17] Sigillito VG, Wing SP, Hutton LV, & Baker KB (1989) Classification of radar returns from the ionosphere using neural networks. Johns Hopkins APL Technical Digest, 10, 262-266

[18] Sharon Summers (1993), Database Source, School of Nursing, University of Kansas Medical Center, Kansas City, KS 66160

[19] Sharon Summers (1993) Source of Post-operative Database: School of Nursing, University of Nursing, University of Kansas Medical Centre, Kansas City, KS 66160 USA.

[20] Database, http://archive.ics.uci.edu/ml/datasets.html, Accessed date 13/01/2009.

[21] Woolery L, Grzymala-Busse J, Summers S, Budihardjo A (1991) The use of machine learning program LERS_LB 2.5 in knowledge acquisition for expert system development in nursing. Computers in Nursing 9, pp. 227-234.

[22] Budihardjo A, Grzymala-Busse J, Woolery L (1991) Program LERS_LB 2.5as a tool for knowledge acquisition in nursing, Proceedings of the 4th Int. Conference on Industrial & Engineering.

[23] Hong, ZQ and Yang JY (1991) Optimal Discriminant Plane for a Small Number of Samples and Design Method of Classifier on the Plane, Pattern Recognition, Vol. 24, No. 4, pp. 317-324.

[24] Gema S, and Gema D, (1984): Stochastic relaxation, Gibbs distributions, and Bayesian restoration of images. IEEE Trans. on Pattern analysis and Machine Intelligence 6:721-741

[25] Uwe D, Peter H, Scheffer T, (2008): Learning from Incomplete Data with Infinite Imputation, 25th Int. conf. on Machine Learning, Helsinki, Finland.

[26] David Williams, Xuejun Liao, and Ya Xue: Incomplete Data Classification using Logistic Regression, Proc. of 22nd Int. Conf. on Machine Learning, Bonn Germany.