Scaling Out Big Data Missing Value Imputations

(Pythia vs. Godzilla)

Christos Anagnostopoulos
School of Computing Science
University of Glasgow, G12 8QQ, Glasgow, UK
christos.anagnostopoulos@glasgow.ac.uk

Peter Triantafillou
School of Computing Science
University of Glasgow, G12 8QQ, Glasgow, UK
peter.triantafillou@glasgow.ac.uk

ABSTRACT

Solving the missing-value (MV) problem with small estimation errors in big data environments is a notoriously resource-demanding task. As datasets and their user community continuously grow, the problem can only be exacerbated. Assume that it is possible to have a single machine ('Godzilla'), which can store the massive dataset and support an ever-growing community submitting *MV imputation* requests. Is it possible to replace Godzilla by employing a large number of cohort machines so that imputations can be performed much faster, engaging cohorts in parallel, each of which accesses much smaller partitions of the original dataset? If so, it would be preferable for obvious performance reasons to access only a subset of all cohorts per imputation. In this case, can we decide swiftly which is the desired subset of cohorts to engage per imputation? But efficiency and scalability is just one key concern! Is it possible to do the above while ensuring comparable or even better than Godzilla's imputation estimation errors? In this paper we derive answers to these fundamentals questions and develop principled methods and a framework which offer large performance speed-ups and better, or comparable, errors to that of Godzilla, independently of which missing-value imputation algorithm is used. Our contributions involve Pythia, a framework and algorithms for providing the answers to the above questions and for engaging the appropriate subset of cohorts per MV imputation request. Pythia functionality rests on two pillars: (i) dataset (partition) signatures, one per cohort, and (ii) similarity notions and algorithms, which can identify the appropriate subset of cohorts to engage. Comprehensive experimentation with real and synthetic datasets showcase our efficiency, scalability, and accuracy claims.

Categories and Subject Descriptors: H. Information Systems; I.5.3 Clustering.

Keywords: Big data; Missing value; Clustering.

1. INTRODUCTION

Data quality is a major concern in big data processing and knowledge management systems. One relevant problem in data quality is the presence of missing values (MVs). The MV problem should be carefully addressed, otherwise bias might be introduced into the induced knowledge. Common solutions to the MV problem either fill-in the MVs (*imputation*) or ignore / exclude them. Imputation entails a *MV substitution algorithm* (MVA) that replaces MVs in a dataset with some plausible values. Imputed data can be treated as reliable as the observed data, but they are as good estimations as the assumptions used to create them.

On the one hand, most computational intelligence and machine learning (ML) techniques (such as neural networks and support vector machines) fail if one or more inputs contains MVs and thus cannot be used for decision-making purposes [1]. Furthermore, the choice of different MVAs affects the performance of ML techniques that are subsequently used with imputed data [2]. On the other hand, the MV problem abounds: it can be found, for instance, in results from medical experimentation and chemical analysis, in datasets from domains such as meteorology and microarray gene monitoring technology [4], and in survey databases [5]. MVs can occur e.g., due to wireless sensor faults, not reacting experiments, or participants skipping survey questions. Industrial and research databases include MVs [6], e.g., maintenance databases have up to 50% of their entries missing [7]. Patient records in medical databases lack some values; interestingly, a database of patients with cystic fibrosis missing more than 60% of its entries was analyzed in [8]. Moreover, gene expression microarray data sets contain MVs, making the need for robust MVAs apparent, since algorithms for gene expression analysis require complete gene array data [9].

Motivations. Given the significance of MVAs, three notes are in order: Firstly, MVAs which can ensure low estimation errors are computationally expensive and typically their performance is largely dependent on dataset sizes. Secondly, nowadays, datasets can be massive. Even worse, existing datasets grow significantly with time; it is not surprising that most MVAs in the literature are typically tested over small-to medium sized datasets. Lastly, as if the scalability limitations imposed by dataset sizes were not enough, in many applications the user community (e.g., in shared scientific datasets in data centers accessed by scientists from all over the world) can be very large and thus the MV imputation input arrival rates can become high as well. These facts pose a scalability nightmare.

The scalability gospel (as established by the seminal work from Google researchers producing the Map-Reduce (MR) [10] data-access paradigm and systems such as the Google

KDD'14, August 24–27, 2014, New York, NY, USA.
Copyright 2014 ACM 978-1-4503-2956-9/14/08 ...$15.00.
http://dx.doi.org/10.1145/2623330.2623615.

File System [11]) rests on the notion of *scaling out*: that is, (i) employ a large number of commodity (off-the-shelf and thus inexpensive) machines, each storing a much smaller partition of the original dataset, and (ii) access them in parallel.

However, MR is not a panacea, for two reasons. First, not all complex problems are 'embarrassingly parallelizable' and amenable to MR techniques. In particular, there exist sophisticated MVAs ensuring small errors, which are not MR-able [12]. Second, in the context of MVAs, even if they were 'embarrassingly parallelizable', not all partitions may be relevant. It may very well be the case that a number of the machines hold data that cannot help (or even hurt) in the MV imputation process. And, obviously, engaging only a fraction of all machines will introduce large benefits: First with respect to performance. MV imputation will be shorter, as these times typically depend on the worst performing machine and with increasing machine numbers the probability of a mall-performing machine increases. Further, overall MV imputation throughput will be higher, as each imputation will be taxing fewer overall system resources (processors, communication bandwidth and disks). Second, with respect to MV estimation errors. In fact, as we shall formally show later, engaging all machines and their dataset partitions may actually introduce large additional MV estimation errors.

Goals. In this work, we will consider a stream of MV inputs (or inputs), i.e., multi-dimensional vectors with some MVs in certain dimensions, arriving at a data system. Typically, the system is presented with a batch of data items with MVs, which must be added to the system after MVs have been estimated. There are two system alternatives. The first is based on employing a single machine which stores the whole of the dataset. We affectionately call this machine *Godzilla*. Godzilla can employ any MVA to perform the MV imputations. As motivated earlier, this approach suffers from several disadvantages. The second alternative employs a (potentially large) number of machines, referred to as *cohorts*, each storing a partition of Godzilla's dataset. Imputation execution engages cohorts in parallel, whereby each cohort runs an MVA on a much smaller local dataset. This can introduce dramatic performance improvements. As an illustration, assuming, say, 50 cohorts and an MVA operating on a dataset of size n with asymptotic complexity $O(n^2)$ (or $O(n^3)$; [3], [4]) a scale-out execution is expected to speedup input processing by a factor of $50^2 = 2,500$ (or $50^3 = 125,000$) as such MVA runs in parallel on a dataset of size $\frac{1}{50}n$. Moreover, this alternative affords the possibility of accessing only a subset of all cohorts for a given input. We will not make any restricting assumptions as to specific characteristics of this system or the method for partitioning the dataset.

The formidable challenges here entail: (i) for data accuracy (estimation-error) reasons, we should ensure that the subset of cohorts contacted achieve similar, if not smaller estimation errors, compared to the errors that Godzilla would yield; (ii) *swiftly* determine the subset of cohorts to engage per imputation, achieving large efficiency/scalability gains.

Contributions. To our knowledge, this is the first study on scaling out MV imputations. We shall derive fundamental knowledge regarding meeting the estimation error and performance goals outlined above. Armed with this knowledge, we shall propose a novel, principally derived framework, *Pythia*, which offers large performance speed-ups and better, or comparable, errors to that of Godzilla given a stream of MV inputs. Pythia's salient contribution is that, given an input (of imputation requests), Pythia is able to predict and engage the appropriate subset of cohorts to employ per imputation. Pythia's prediction process relies on (i) the concept of per cohort-dataset *signature*, which derives from the (local) dataset of a cohort and (ii) novel similarity notions and algorithms which, based on each imputation request and cohort signatures, can determine the best subset of cohorts to engage. Finally, we will provide comprehensive experimental evidence substantiating and showcasing Pythia's accuracy and performance, using a variety of metrics and real and synthetic datasets.

The paper is structured as follows: Section 2 reports on background and discusses related work, while Section 3 introduces the problem fundamentals of scaling out the MV imputations. In Section 4 and Section 5 we introduce the Pythia framework and propose two schemes. In Section 6 we evaluate our framework and Section 7 concludes the paper.

2. BACKGROUND & RELATED WORK

2.1 Missing data

Assume a data set $\mathcal{X}$ of d-dimensional data points with some MVs on a certain dimension X_i. Data on X_i are said to be *missing completely at random* (MCAR) if the probability of MV on X_i, q, is unrelated to the value of X_i itself or to the values of any other dimensions. If data are MCAR, a reduced sample of $\mathcal{X}$ will be a random sub-sample of $\mathcal{X}$; MCAR assumes that the distributions of MVs and complete data are the same. Data on X_i are said to be *missing at random* (MAR) if q depends on the observed data, but does not depend on the MV itself. In MAR, the dimension associated with MVs has a relation to other dimensions, i.e., MVs can be estimated by using the complete data of other dimensions. It is impossible to test whether the MAR condition is satisfied for $\mathcal{X}$ because, since the (actual) values of missing data are not known, we cannot compare the values of those with and without missing data to see if they differ systematically on that X_i. Data on X_i are *missing not at random* (MNAR) if q depends on the MVs and, thus, missing data cannot be estimated by using the existing dimensions; MNAR is rarely applicable in practice.

2.2 Related work

Missing data hinder the application of many statistical analysis and ML techniques available in off-the-shelf software. To analyze $\mathcal{X}$ with MVs, certain MVAs have been proposed [13]. The simplest method is discarding the data points with MVs or removing the corresponding dimensions. Both removals of such points and dimensions result in decreasing the information content of $\mathcal{X}$ and are applicable only when (i) $\mathcal{X}$ contains a small amount of MVs, and (ii) the analysis of the remaining complete points will not be biased by the removal. There are many MVAs varying from naïve methods, e.g., mean imputation, to some more robust methods based on relationships among dimensions. In the *dummy variable adjustment*, MVs are set to some arbitrary value. The *mean / mode imputation* replaces MVs of a dimension by the sample mean / mode of all observed values of that dimension. In *hot deck* MVA [14], a MV is filled in with a value from an estimated distribution w.r.t. $\mathcal{X}$. In the K-nearest neighbors MVA [15], the MVs of a point are imputed considering the K most similar (observed) points from $\mathcal{X}$. The regression- and likelihood-based MVAs are introduced in [16]. In *regression-based imputation* [17], the MVs of a point are estimated by regression of the dimensions corresponding to MVs on the dimensions associated to the observed values of that point. This approach argues that dimensions have relationships among themselves; if no relationships exist among dimensions in $\mathcal{X}$ and the dimensions corresponding to MVs, such MVA will not be precise for imputation. *Likelihood-based imputation* [16] is based on parameter estimation in the presence of MVs, i.e., $\mathcal{X}$'s parameters are estimated by maximum likelihood or maximum a posteriori procedures relying on variants of the Expectation-Maximization algorithm. The *multiple imputation* MVA [18], instead of filling in a single value for each MV, replaces each MV with a set of plausible values that represent the uncertainty about the actual value to impute. These multiply-imputed datasets are then analyzed by using standard procedures for complete data and combining the results from these analyses. In case of MVs in time series, the models in [19] (using dynamic Bayesian networks), [20] (using matrix completion), and [21] (using Gaussian mixtures clustering) recover MVs in motion capture sequences, vital signs, and micro-array gene expression streams, respectively. Furthermore, ML-based MVAs, e.g., decision-trees and rule-based methods, generate a model from $\mathcal{X}$ that contain MVs, which is used to perform classification that imputes the MVs (see [2] and the references therein). Finally, the imputation framework [6] applies most existing MVAs (base methods) to improve their accuracy of imputation while preserving the asymptotic computational complexity of the base methods. The interested reader could also refer to [6], [9] and [22] (and the references therein) for a comprehensive survey of the most recent MVAs.

3. PROBLEM ANALYSIS & FUNDAMENTALS

3.1 Definitions & Notations

Definition 1. Given a set $\mathcal{X}$ of d-dimensional data points, $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{|\mathcal{X}|}\}$, for each $\mathbf{x}_i$ we define $\mathbf{w}_i = [w_{ik}]^\top$ with $w_{ik} = 0$ whenever $\mathbf{x}_i$'s k-th dimensional value is missing; otherwise $w_{ik} = 1$. We express $\mathbf{x}_i$ as $(\mathbf{z}_i, \mathbf{z}_i^{\mathbf{m}})$, where $\mathbf{z}_i \in \mathbb{R}^n$ denotes observed values and $\mathbf{z}_i^{\mathbf{m}} \in \mathbb{R}^{(d-n)}$ denotes MVs, with $n = \sum_{k=1}^{d} w_{ik}$.

Definition 2. Given a finite integer $m > 0$, $\mathcal{X}_i$ is a partition of $\mathcal{X}$ such that $\mathcal{X} \equiv \cup_{i=1}^{m} \mathcal{X}_i$ and $\mathcal{X}_i \neq \mathcal{X}_j, i \neq j$. S_i denotes the machine (*cohort*), which maintains $\mathcal{X}_i$, performs a MVA over $\mathcal{X}_i$, and is indexed by i, $i = 1, \ldots, m$. $\mathcal{S} = \{S_i\}_{i=1}^{m}$ is the set of all cohorts. The (imaginary) *Godzilla* S_0 assembles all $\mathcal{X}_i$ and is capable of performing a MVA over $\mathcal{X}$.

Definition 3. A single MV input on MVA is $\mathbf{i} = (\mathbf{x}, \mathbf{w})$ and output is $\hat{\mathbf{x}}$ expressed by $(\mathbf{z}, \hat{\mathbf{z}}^{\mathbf{m}})$. $\hat{\mathbf{x}} \in \mathbb{R}^d$ is referred to as *estimate* containing $\hat{\mathbf{z}}^{\mathbf{m}} \in \mathbb{R}^{(d-n)}$ of imputed MVs by MVA. If $\mathbf{x}_a$ is the actual vector, the absolute reconstruction error is $e = \| \hat{\mathbf{x}} - \mathbf{x}_a \|$; $\| \mathbf{x} \|$ denotes the Euclidean norm.

3.2 MVAs in our framework

As our contributions are independent of any particular MVA, we overview two popular and representative MVAs as would be used in our framework. To exemplify our framework and methods, we employ the weighted K-nearest neighbors (KNN) [15] and sequential multivariate regression imputation method (REG) [17]. These MVAs are widely used for multivariate imputation in many scientific areas.

3.2.1 Weighted K-nearest neighbors imputation

KNN is widely used [22] since it has many attractive characteristics: it is a non-parametric method, which does not require the creation of a predictive model for each dimension with MV and takes into account the correlation structure of the data. KNN is based on the assumption that points close in distance are potentially similar. For given input $(\mathbf{x}_i, \mathbf{w}_i)$ with $\mathbf{x}_i = (\mathbf{z}_i, \mathbf{z}_i^{\mathbf{m}})$, KNN calculates a weighted Euclidean distance D_{ij} between $\mathbf{x}_i$ and $\mathbf{x}_j \in \mathcal{X}$ such that

$$D_{ij} = \left(\frac{\sum_{k=1}^{d} w_{ik} w_{jk} (x_{ik} - x_{jk})^2}{\sum_{k=1}^{d} w_{ik} w_{jk}} \right)^{1/2} .$$

The MV of the k-th dimension of $\mathbf{x}_i$ (i.e., $z_{ik}^{\mathbf{m}}$ of $\mathbf{z}_i^{\mathbf{m}}$) is estimated by the weighted average of non-MVs of the K most similar $\mathbf{x}_j$ to $\mathbf{x}_i$, i.e., $\hat{z}_{ik}^{\mathbf{m}} = \sum_{j=1}^{K} \frac{D_{ij}^{-1}}{\sum_{v=1}^{K} D_{iv}^{-1}} x_{jk}$. KNN is typically used with K=10,15,20; theses values have been favored in previous studies [22], [23]. (In our experiments we will use K=10).

3.2.2 Sequential multivariate regression imputation

REG estimates the MVs by fitting a sequence of regression models and drawing values from the corresponding predictive distributions. Let $Y_1, \ldots, Y_{d-n}$ denote $d-n$ (dependent) variables with MVs, sorted in ascending order to the number of MVs and $\mathbf{X} = [X_1, \ldots, X_n]^\top$ denote n (predictor) variables with no MVs. REG consists of c rounds. In round 1, step 1, we regress the variable with the fewest number of MVs, Y_1, on $\mathbf{X}$ imputing the MVs under the appropriate regression model; e.g., if Y_1 is continuous, categorical, or binary then ordinary least squares, generalized logit, or logistic linear regression is applied, respectively. In step 2, after estimating the regression coefficients β of the model from step 1, we use the estimated $\hat{\beta}$ to impute the MVs of Y_1. In step 3, we update $\mathbf{X}$ by appending Y_1 and continue to variable, say Y_2, with the next fewest MVs and repeat the process using updated $\mathbf{X}$ as predictors until all the variables have been imputed. That is, Y_1 is regressed on $\mathbf{U} = \mathbf{X}$; Y_2 is regressed on $\mathbf{U} = (\mathbf{X}, Y_1)$, where Y_1 has imputed MVs; Y_3 is regressed on $\mathbf{U} = (\mathbf{X}, Y_1, Y_2)$, where Y_1 and Y_2 have imputed MVs, and so on. Steps 1 to 3 are then repeated in rounds 2 through c, modifying the predictors set to include all Ys except the one used as the dependent variable. Hence, regress Y_1 on $\mathbf{X}$ and $Y_2, \ldots, Y_{d-n}$; regress Y_2 on $\mathbf{X}$ and $Y_1, Y_3, \ldots, Y_{d-n}$, and so on. Repeated cycles continue for c rounds, or until stable imputed MVs occur.

3.3 On Cohort vs. Godzilla errors

We consider a discrete time domain $t \in \mathbb{T}$ and at instance $t = 1, 2, \ldots$, we are given input $\mathbf{i}[t]$. Assume that Godzilla S_0 exists and is capable of invoking a certain MVA for $\mathbf{i}[t]$. At first thought, one could claim that, since Godzilla has global knowledge (i.e., the union of all $\mathcal{X}_i$), the corresponding estimate $\hat{\mathbf{x}}_G[t]$ would be *better* (in terms of reconstruction error $e_G[t]$) than $\hat{\mathbf{x}}_i[t]$ of each S_i (with error $e_i[t]$). However, this does not always hold true. It depends on the probability density function (pdf) of $\{\mathcal{X}_i\}$ and the (possibly unknown) pdf of $\mathbf{z}[t]$, $\mathbf{z}^{\mathbf{m}}[t]$, and $\mathbf{w}[t]$.

THEOREM 1. *Let e_G and e_i denote the estimate error of Godzilla S_0 and cohort S_i. It is not always true that $e_G < e_i, \forall S_i \in \mathcal{S}$.*

PROOF. To prove Theorem 1, suppose its converse were true. Then it suffices to show counterexamples. Consider the mean imputation (MEAN) and the KNN. Consider that points in $\mathcal{X}_i$ are normally distributed, $\mathcal{N}(\mu_i, \sigma_i^2)$, with mean μ_i and variance σ_i^2 and $|\mu_i - \mu_j| >> 0, i \neq j$. Evidently, S_0's data set $\mathcal{X} = \cup_{i=1}^{m} \mathcal{X}_i$ follows the mixture $\mathcal{N}(\mu, \sigma^2)$ with $\mu = \sum_{i=1}^{m} a_i \mu_i$, $\sigma^2 = \sum_{i=1}^{m} a_i((\mu_i - \mu)^2 + \sigma_i^2)$; $a_i > 0, \sum_{i=1}^{m} a_i = 1$. If we were told that all (both observed $\mathbf{z}$ and unobserved $\mathbf{z}^{\mathbf{m}}$) inputs followed $\mathcal{N}(\mu_j, \sigma_j^2)$ for some $j, 1 \leq j \leq m$ then we should have engaged only S_j thus yielding $e_j < e_G$ in case of MEAN, and $e_j = e_G$ in case of KNN (for $K << |\mathcal{X}_j|$) and avoiding engaging all S_i. □

Furthermore, consider that all $\mathcal{X}_i$ follow exactly the same distribution; consequently, S_0's $\mathcal{X}$ follows the same distribution. Then, regardless of any knowledge on the pdfs of inputs, we could randomly select one cohort from $\mathcal{S}$, thus, yielding $e_i = e_G, \forall S_i \in \mathcal{S}$, and avoiding engaging all cohorts.

Example 1: Consider $m = 3$ cohorts S_1, S_2, S_3 with 2D datasets $\mathcal{X}_i$, corresponding joint pdfs f_1, f_2, f_3 and a Godzilla S_0 with $\mathcal{X} = \cup_{i=1}^{3} \mathcal{X}_i$ whose joint pdf f_G is shown in Fig. 1(a). Assume REG, KNN, and MEAN MVAs. We are given a stream of 10^4 inputs $\mathbf{i}[1], \ldots, \mathbf{i}[t]$ and assume that we *know* the pdf of each $\mathbf{i}[t]$, i.e., its observed and MVs are known to be produced either by f_1, f_2, or f_3. For each $\mathbf{i}[t]$, we invoke a MVA (a) on S_0 and obtain $e_G[t]$, (b) only on the cohort S_j with the same pdf f_j as that of the input and obtain $e_j[t]$, and (c) on all cohorts, aggregate their estimates by taking their average and obtain $e_{all}[t]$. Fig. 1(b) shows the root-mean-square error (RMSE) e_G, e_j, and e_{all} for all MVAs. We observe that the knowledge of the pdf of each input results to a significantly lower error e_j, because we engage only the cohort S_j corresponding to the same pdf as that of the input. Godzilla produces a relatively high e_G (for all MVAs) with high computational cost due to processing high volumes of data. Moreover, the parallel execution of MVAs over all cohorts for each input produces a high e_{all}. Unfortunately, the pdf of an incoming input is not known, especially, the pdf of the MVs is unknown since they are never observed. Moreover, we can achieve high parallelism with concurrently engaging all cohorts but, we also obtain high error, because there might be a subset of cohorts that *adversely* contribute to the aggregated estimate, e.g., due to the fact that the corresponding pdfs of their data sets are different from those of the inputs (see Example 2). Note, however, that in the case of MEAN, $e_G = e_{all}$.

3.4 On computing good cohort subsets

Here we show: (i) that computing the best cohorts subset is computationally hard, (ii) that even if an efficient heuristic can be found, it would not be desirable for our purpose since it would require communication with all cohorts, hence, another approach is needed, like our signature-based prediction approach and (iii) that as exemplified using our reference popular MVAs, it is highly beneficial to engage only a good cohort subset per imputation. The above showcases thus the traits and benefits of our approach.

In our framework, we utilize a node called Pythia that attempts to predict the best cohorts subset per input. Pythia receives input $\mathbf{i}[t] = (\mathbf{x}[t], \mathbf{w}[t])$ with $0 < n[t] = \sum_{k=1}^{d} w_k[t] < d$. In the remainder, the time index t is omitted for the sake of readability. Of course, Pythia can, trivially, engage all cohorts in parallel. Each cohort S_i locally produces an estimate $\hat{\mathbf{x}}_i$ (through MVA invocation) and provides it to Pythia. Then, Pythia takes their average value $\hat{\mathbf{x}} = \frac{1}{m} \sum_{i=1}^{m} \hat{\mathbf{x}}_i$. Let us denote such method as the *All Cohorts Method*, notated by ACM, so to differentiate it from Pythia's sophisticated methods. ACM implies that all cohorts are equal candidates and available for providing an estimate. It would have been preferable if Pythia could engage a subset $\mathcal{S}' \subset \mathcal{S}$ of cohorts whose average estimate $\hat{\mathbf{x}}' = \frac{1}{|\mathcal{S}'|} \sum_{S_i \in \mathcal{S}'} \hat{\mathbf{x}}_i$ would be equal to $\hat{\mathbf{x}}$, or more interestingly, if Pythia could engage the minimum subset of cohorts whose average estimate is close to $\hat{\mathbf{x}}$ for each input.

Determining the minimum cohorts subset whose aggregate estimate is close to $\hat{\mathbf{x}}$ calls to mind the Subset Sum Problem (SSP) [24]: Consider a pair $(\mathcal{I}, s)$, where $\mathcal{I}$ is a set of $m > 0$ positive integers and s is a positive integer. SSP asks for a subset of $\mathcal{I}$ whose sum is closest to, but not greater than, s. SSP is NP-hard [24]. Consider now the following problem, referred to as Minimum Subset Average Problem (MSAP).

Problem 1. (MSAP) Given $(\mathcal{I}, s)$, find the minimum subset $\mathcal{I}'$ with average s' subject to $\lfloor s' \rfloor = s$ or $\lceil s' \rceil = s$ (C1).

THEOREM 2. *MSAP is NP-hard.*

PROOF. If there is a polynomial-time algorithm for MSAP, then a polynomial-time algorithm can be developed for SSP. Assume there exists a polynomial algorithm $A(\mathcal{I}, s)$ that solves MSAP, i.e., $A(\mathcal{I}, s)$ finds in polynomial time the minimum subset $\mathcal{I}'$ subject to constraint C1 in Problem 1. Then, $A(\mathcal{I}, s)$ can be used to solve SSP with $(\mathcal{I}, ms)$, $m = |\mathcal{I}|$. In general, any solution $B(\mathcal{I}, s)$ of SSP with $(\mathcal{I}, s)$ can be formulated as Algorithm 1. If the complexity of $A(\mathcal{I}, s)$ is a polynomial $\mathcal{Q}(m)$ then the complexity of $B(\mathcal{I}, s)$ is $O(m\mathcal{Q}(m))$. But, this implies that there is a polynomial-time algorithm for SSP. Hence, no polynomial-time algorithm exists for MSAP. □

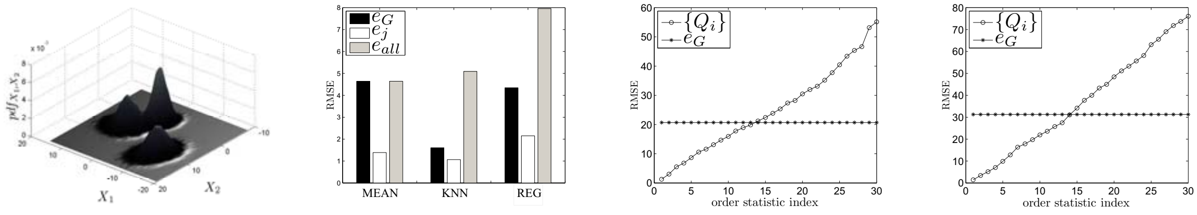
ALGORITHM 1: $B(\mathcal{I}, s)$

```
Input: $\mathcal{I}, s$
Output: $\mathcal{I}'$
for $1 \leq k \leq |\mathcal{I}|$ do
    call $A(\mathcal{I}, \frac{s}{k})$;
    If a subset $\mathcal{I}'$ of $\mathcal{I}$ with $k$ elements is found, whose
    elements have an average $k'$ such that $\lfloor k' \rfloor = s/k$ or
    $\lceil k' \rceil = s/k$ Then return $\mathcal{I}'$
end
```

THEOREM 3. *Given input* $\mathbf{i}$*, the problem of finding the minimum subset $\mathcal{S}' \subset \mathcal{S}$ of cohorts, whose average estimate $\hat{\mathbf{x}}'$ gives the same reconstruction error as $\hat{\mathbf{x}}$ is NP-hard.*

PROOF. Let $e = \parallel \hat{\mathbf{x}} - \mathbf{x}_a \parallel$ and $e' = \parallel \hat{\mathbf{x}}' - \mathbf{x}_a \parallel$. In order to show that the problem of finding the minimum subset $\mathcal{S}'$ with $e' = e$ is NP-hard, it suffices to show that finding the minimum subset $\mathcal{S}' \subset \mathcal{S}$ of cohorts such that $\parallel \hat{\mathbf{x}}' \parallel = \parallel \hat{\mathbf{x}} \parallel$ subject to C1 is NP-hard. Consider the set $\mathcal{I}^0 = \{\lfloor \parallel \hat{\mathbf{x}}_i \parallel \rfloor\}_{i=1}^{m}$, and $\mathcal{I}^1 = \{\lceil \parallel \hat{\mathbf{x}}_i \parallel \rceil\}_{i=1}^{m}$, $\parallel \hat{\mathbf{x}}_i \parallel > 0, \forall i$. Since MSAP, which deals with integers is NP-hard from Theorem 2, MSAP with $(\mathcal{I}^0, \lfloor \parallel \hat{\mathbf{x}} \parallel \rfloor)$ and $(\mathcal{I}^1, \lceil \parallel \hat{\mathbf{x}} \parallel \rceil)$ is also NP-hard. □

SSP and MSAP are NP-hard, however, one is often satisfied with an approximate, sub-optimal solution, i.e., in polynomial time; see [25] for SSP. Nevertheless, even if Pythia



(a) Joint pdf $f_G(X_1, X_2)$

(b) RMSE using MEAN, KNN, REG

(c) RMSE using KNN

(d) RMSE using REG

Figure 1: (a) Joint pdf; (b) RMSE e_G, e_j, e_{all} using MEAN, KNN and REG for $m = 3$ in Example 1; (c-d) RMSE of Godzilla and order statistics Q_i of ACM using KNN and REG for $m = 30$ in Example 2.

were able to use such heuristic to find the minimum set $\mathcal{S}'$ for given input (let m be small) then this would still not be preferable given our goals. That is because, in order to obtain $\mathcal{S}'$ for a given input, Pythia would *firstly* have to engage all cohorts and consequently, based on their estimates, produce $\mathcal{S}'$. What we want is for Pythia to guess/predict the most appropriate $\mathcal{S}'$, which gives the same or, hopefully, smaller reconstruction error than that of $\mathcal{S}$ *without having to access all cohorts*! For instance, this guess can be interpreted as follows: cohort $S_i \in \mathcal{S}$ might consider $\mathbf{z}$ (of input $\mathbf{i}$) as an observation which is deemed *unlikely* w.r.t. $\mathcal{X}_i$. Based on the fact that a MVA highly depends on $\mathcal{X}_i$, S_i will probably provide a bad estimate for $\mathbf{i}$ (w.r.t. e_i). Were Pythia capable of predicting the *unsuitability* of S_i providing a good estimate *before* engaging S_i then Pythia could have excluded S_i from $\mathcal{S}'$.

The task of predicting $\mathcal{S}'$ per input involves the following issues: (a) the joint pdf of the MVs is evidently unknown since the actual values of $\mathbf{z^m}$ are not observed; (b) it is not feasible to identify the joint pdf that generates $\mathbf{z}$, since we have only one sample from this at a time; (c) it is not suitable to assume that $\mathbf{z}$ is produced by a certain pdf at time t, which remains also the same for subsequent $\mathbf{z}[\tau], \tau > t$. This is getting more difficult when dealing with non-stationary distributions of $\mathbf{z}$ and $\mathbf{w}$, which is not a rare situation.

Example 2: Consider $m = 30$ cohorts. We are given a stream of 10^4 inputs where the joint pdf of each input is unknown. For each input, we invoke a MVA (KNN and REG) on Godzilla and on all cohorts in parallel, and aggregate their estimates (ACM). For each input, we obtain the order statistics $Q_1 = \min_i\{e_i\}, \ldots, Q_{30} = \max_i\{e_i\}$ of the corresponding errors of all cohorts and plot their average values in Fig. 1(c-d); the e_G is shown for comparison. We can observe that more than 40% of cohorts provide lower error to that of Godzilla for KNN and REG. This indicates that it is of high importance to predict such subset of cohorts for each input while knowing neither the pdfs of the cohorts' sets nor the pdf of each input. Note that ACM in this case produces a higher average error than even Godzilla. Furthermore, we observe that for each input there is an *ideal* cohort that gives the minimum error; note that $\bar{Q}_1$ is 93% / 95% smaller than e_G for KNN / REG. An *ideal* Pythia has to predict $\mathcal{S}'$ hopefully including the ideal cohort and/or those S_i with $e_i < e_G$ for each input. We now formulate our problems.

Problem 2. Determine what *information* each $S_i \in \mathcal{S}$ a-priori must convey to Pythia in order to predict whether S_i is suitable for providing a (local) good estimate $\hat{\mathbf{x}}_i$ given an input, i.e., whether S_i should be a member of $\mathcal{S}'$. This information is referred to as the *signature* P_i of $\mathcal{X}_i$.

Problem 3. Determine how signatures $\{P_i\}_{i=1}^m$ are updated for each input.

4. THE PYTHIA FRAMEWORK

Pythia aims to solve the above problems. Predicting $\mathcal{S}'$ for each input, based on per-cohort signatures, avoids the fundamental problems of NP-hardness of exact solutions and of the need to on-the-fly engage all cohorts for approximate heuristic solutions.

Each cohort S_i constructs a signature P_i from $\mathcal{X}_i$. P_i reflects the current structure of data points in $\mathcal{X}_i$. The idea behind a signature is that S_i is engaged for a given $\mathbf{i}$ once $\mathbf{x}$ can be 'explained' through P_i. S_i provides its (locally) created P_i to Pythia, which stores all signatures forming $\mathcal{P} = \{P_i\}_{i=1}^m$. Figure 2(a) pictorially depicts the framework's operation. The operation of the framework is as follows: Given $\mathbf{i}$,

1. Pythia predicts $\mathcal{S}' \subseteq \mathcal{S}$ w.r.t. $\mathcal{P}$
2. Pythia then engages only the cohorts from S' sending $\mathbf{i}$ to them.
3. Each $S_i \in \mathcal{S}'$
   (a) invokes a MVA and
   (b) provides its estimate $\hat{\mathbf{x}}_i$ to Pythia.
4. Pythia constructs the aggregate estimate $\hat{\mathbf{x}}$ that is sent to cohorts from $\mathcal{S}'$.
5. Each $S_i \in \mathcal{S}'$ can exploit $\hat{\mathbf{x}}$ for updating its P_i.
6. Pythia uses $\hat{\mathbf{x}}$ for updating $\mathcal{P}$.

4.1 Signatures

In this work, P_i refers to a clustering structure over $\mathcal{X}_i$ providing a set of representative points (clusters) $\mathcal{C}_i$. Each cohort $S_i \in \mathcal{S}$ employs the Adaptive Resonance Theory (ART) [26], an unsupervised learning model from the competitive learning paradigm, in order to locally construct P_i over $\mathcal{X}_i$. In ART, whose algorithm is shown as Algorithm 2, each $\mathbf{x}_k \in \mathcal{X}_i$ is processed by finding the nearest cluster $\mathbf{c}^* \in \mathbb{R}^d$ to $\mathbf{x}_k$, i.e., $\mathbf{c}^* = \arg\min_{\mathbf{c} \in \mathcal{C}_i} \parallel \mathbf{c} - \mathbf{x}_k \parallel$, where $\mathcal{C}_i$ is the set of clusters. Then, it is allowed $\mathbf{x}_k$ to modify/update $\mathbf{c}^*$ only if $\mathbf{c}^*$ is sufficiently close to $\mathbf{x}_k$ ($\mathbf{c}^*$ is said to 'resonate' with $\mathbf{x}_k$) i.e., if $\parallel \mathbf{c}^* - \mathbf{x}_k \parallel \leq \rho_i$ for some *vigilance* $\rho_i > 0$. In this case, $\mathbf{c}^*$ is updated through the rule $\mathbf{c}^* \leftarrow \mathbf{c}^* + \eta_i(\mathbf{x}_k - \mathbf{c}^*)$, where $\eta_i \in (0, 1)$ is a learning rate, which gradually decreases. Otherwise, i.e., $\parallel \mathbf{c}^* - \mathbf{x}_k \parallel > \rho_i$, a new cluster $\mathbf{c}$ is formed handling $\mathbf{x}_k$ such that $\mathbf{c} = \mathbf{x}_k$ and $\mathcal{C}_i \leftarrow \mathcal{C}_i \cup \{\mathbf{c}\}$.

Definition 4. Cohort S_i's signature P_i over $\mathcal{X}_i$ is the triple

$$P_i = \langle \mathcal{C}_i, \rho_i, \eta_i \rangle. \tag{1}$$

ALGORITHM 2: ART algorithm at cohort S_i

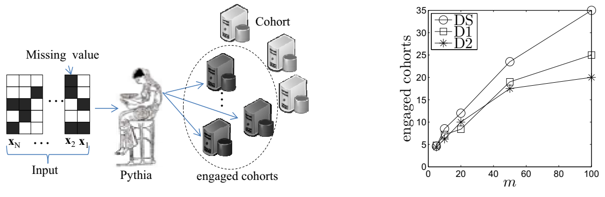
```
Input: $\mathcal{X}_i, \eta_i, \rho_i$
Output: $\mathcal{C}_i$
$\mathcal{C}_i = \{\mathbf{x}_1\}$;
for $1 < k \leq |\mathcal{X}_i|$ do
    $b^* = \| \mathbf{c}^* - \mathbf{x}_k \| = \min_{\mathbf{c} \in \mathcal{C}_i} \| \mathbf{c} - \mathbf{x}_k \|$;
    if $b^* > \rho_i$ then
        $\mathcal{C}_i \leftarrow \mathcal{C}_i \cup \{\mathbf{x}_k\}$;
    else
        $\mathbf{c}^* \leftarrow \mathbf{c}^* + \eta_i(\mathbf{x}_k - \mathbf{c}^*)$;
    end
end
```

Definition 5. We say that $\mathbf{x}$ is a *member* of P_i, notated $\mathbf{x} \in P_i$, iff $\min_{\mathbf{c} \in \mathcal{C}_i} \| \mathbf{c} - \mathbf{x} \| \leq \rho_i$; otherwise, $\mathbf{x} \notin P_i$.

The statement '$\mathbf{x} \in P_i$' denotes that there is at least one $\mathbf{c} \in \mathcal{C}_i$ such that $\mathbf{x}$ is placed close to $\mathbf{c}$ with distance less than ρ_i, for instance, the closest cluster $\mathbf{c}^*$ to $\mathbf{x}$. The more clusters $\mathbf{c} \in \mathcal{C}_i$ satisfy the criterion $\| \mathbf{c} - \mathbf{x} \| \leq \rho_i$, the more appropriate $\mathcal{C}_i$ is for $\mathbf{x}$. In this sense, if $\mathbf{x} \in P_i$ then $\mathbf{x}$ can be represented by at least one cluster from $\mathcal{X}_i$. Based on this intuition, if $\mathbf{x} \in P_i$, cohort S_i provides a rather good estimate for some missing parts of $\mathbf{x}$ compared to a cohort S_j associated with a P_j for which it holds true that $\mathbf{x} \notin P_j$. The latter case indicates that no cluster from $\mathcal{C}_j$ can be a representative point for $\mathbf{x}$.

Since ρ_i represents a threshold of similarity between points and clusters, thus, guiding ART in determining when a new cluster should be formed, it should depend on $\mathcal{X}_i$. In order to give a physical meaning to ρ_i, it is expressed through a set of percentages $\alpha_k \in (0, 1)$ of the ranges between the lowest $x_k^{\min}$ and highest $x_k^{\max}$ values of each dimension k of points in $\mathcal{X}_i$, $k = 1, \ldots, d$. Let $\mathbf{r}_i = [(x_1^{\max} - x_1^{\min}), \ldots, (x_d^{\max} - x_d^{\min})]^\top$ and the diagonal $d \times d$ matrix $\mathbf{A}$ with $\mathbf{A}[k, k] = \alpha_k$. Then $\rho_i = \| \mathbf{A}\mathbf{r}_i \|$. High α_k values result to a low number of clusters and vice versa. Each S_i determines a ρ_i over $\mathcal{X}_i$, creates P_i through Algorithm 2, and sends P_i to Pythia.

Note: when dealing with mixed-type data points, e.g., consisting of categorical, binary, and continuous attributes, we can adopt appropriate distance metrics [27] for the distance between $\mathbf{x}_k$ and $\mathbf{x}_l$ instead of using the Euclidean distance $\| \mathbf{x}_k - \mathbf{x}_l \|$; this does not spoil the generality of signature creation.



(a) Pythia

(b) Number of engaged cohorts

Figure 2: (a) Inputs with MVs, Pythia and engaged cohorts; (b) Engaged cohorts against m (COE).

4.2 Cohort prediction schemes

Up to this point, we have shown how to use signatures as a guiding light to select appropriate cohorts for MV imputations. Now, our concern is twofold: MV imputations must be (i) low cost and (ii) high accuracy. Low cost (once signature processing is performed) refers to the communication cost between Pythia and cohorts and to the cost of running MVAs at cohorts. High accuracy refers to low RMSE. Therefore, we present algorithms with these in mind.

4.2.1 *Cost-aware algorithm: Top-$\mathcal{K}$ Cohort scheme*

For simplicity we present the top-1 (Best) Cohort (BC) scheme, i.e., $\mathcal{K} = 1$. Pythia is not involved in producing the (final) estimate $\hat{\mathbf{x}}$, instead, only one cohort (best cohort) is engaged for doing this locally. Pythia communicates only with the best cohort, which runs the MVA, thus, this optimizes our cost metric. Given $\mathbf{i}$, Pythia determines the best cohort $S^* \in \mathcal{S}$ with $P^* = \langle \mathcal{C}^*, \rho^*, \eta^* \rangle$ such that (**A1**) $\mathbf{c}^* = \arg\min_{\mathbf{c} \in \cup_{i=1}^m \mathcal{C}_i} \| \mathbf{c} - \mathbf{z} \|$ and $\mathbf{c}^* = \arg\min_{\mathbf{c} \in \mathcal{C}^*} \| \mathbf{c} - \mathbf{z} \|$, i.e., $\mathbf{c}^* \in \mathcal{C}^*$ is the closest cluster to $\mathbf{z}$ among all clusters from all signatures, and (**A2**) $\mathbf{z} \in P^*$. Note that $\mathbf{z} \in \mathbb{R}^n$ with $0 < n = \sum_{k=1}^d w_k < d$ provided that $\mathbf{x}$ contains $d - n$ MVs. In order to evaluate '$\mathbf{z} \in P^*$' Pythia calculates $\rho^{*(n)} \leq \rho^*$ associated with the n dimensions of $\mathbf{r}^*$ corresponding to the n non-MVs. Then, it checks if $\| \mathbf{c}^* - \mathbf{z} \| \leq \rho^{*(n)}$ dealing only with the n dimensions of $\mathbf{c}^*$. Pythia engages only S^*, which produces the final $\hat{\mathbf{x}}$. If there is no cohort that satisfies criteria A1 and A2, BC engages the cohort that satisfies only criterion A1. If $\mathcal{K} > 1$ one can repeat the above criteria for the top $\mathcal{K}$ cohorts ranked with the distance between the corresponding $\mathbf{c}_j^*$ and $\mathbf{z}$, $1 \leq j \leq \mathcal{K} < m$. In this case the final $\hat{\mathbf{x}}$ is produced by aggregating all $\hat{\mathbf{x}}_j$.

4.2.2 *Accuracy-aware algorithm: Cohorts Outlier Elimination scheme*

Cohorts Outlier Elimination (COE) trades off additional cost for improving our other metric, accuracy. Given $\mathbf{i}$, Pythia checks whether $\mathbf{z} \in P_i$. This is achieved once Pythia, for each cohort S_i, calculates $\rho_i^{(n)} \leq \rho_i$ associated with the n dimensions of $\mathbf{r}_i$ corresponding to the n non-MVs. If $\| \mathbf{c}_i^* - \mathbf{z} \| \leq \rho_i^{(n)}$ (dealing only with the n dimensions of $\mathbf{c}^*$) with $\mathbf{c}_i^* = \arg\min_{\mathbf{c} \in \mathcal{C}_i} \| \mathbf{c} - \mathbf{z} \|$ then $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{S_i\}$. Once $\mathcal{S}'$ is determined with $\ell = |\mathcal{S}'| \leq |\mathcal{S}| = m$, Pythia engages only cohorts from $\mathcal{S}'$ and obtains their corresponding estimates $\hat{\mathbf{x}}_i, i = 1, \ldots, \ell$. The aggregate estimate $\hat{\mathbf{x}}$ determined by Pythia is

$$\hat{\mathbf{x}} = \sum_{S_i \in \mathcal{S}''} \hat{\mathbf{x}}_i b_i \ , \ b_i = \frac{\| \mathbf{z} - \mathbf{c}_i^* \|^{-1}}{\sum\limits_{S_j \in \mathcal{S}''} \| \mathbf{z} - \mathbf{c}_j^* \|^{-1}}, \qquad (2)$$

where b_i is the weight for estimate $\hat{\mathbf{x}}_i$ normalized by the sum of inverse distance from the closest cluster $\mathbf{c}_i^*$ to $\mathbf{z}$ from cohort $S_i \in \mathcal{S}''$. The set $\mathcal{S}'' \subseteq \mathcal{S}'$ contains cohorts $S_i \in \mathcal{S}'$ whose estimates are not considered *outliers* in $\mathcal{E} = \{\| \hat{\mathbf{x}}_1 \|, \ldots, \| \hat{\mathbf{x}}_\ell \|\}$. This is achieved by computing the statistic

$$u_{i,\mathcal{E}} = \frac{| \| \hat{\mathbf{x}}_i \| - median(\mathcal{E})|}{mad(\mathcal{E})} \qquad (3)$$

for each $\| \hat{\mathbf{x}}_i \| \in \mathcal{E}$ and then considering $\hat{\mathbf{x}}_i$ as outlier if $u_{i,\mathcal{E}}$ exceeds a certain cutoff, usually 2.5 or 3.0 [28]. The $median(\mathcal{E})$ and $mad(\mathcal{E})$ is the sample median and median absolute deviation about the median of $\mathcal{E}$, respectively. Pythia provides $\hat{\mathbf{x}}$ to each $S_i \in \mathcal{S}''$ for updating their signatures; see Section 5.1. If $\mathcal{S}' = \emptyset$, Pythia engages all cohorts; if $\mathcal{S}'' = \emptyset$, Pythia engages all cohorts from $\mathcal{S}'$.

4.3 Pythia asymptotic complexity

In COE, given $\mathbf{i}$ Pythia evaluates '$\mathbf{z} \in P_i$', $\forall P_i \in \mathcal{P}$, i.e., it performs one nearest neighbor (1NN) search for each P_i over $\mathcal{C}_i$. We adopt a d-dimensional tree structure [31] for each P_i over the clusters of $\mathcal{C}_i$. Let $\xi = \frac{1}{m}\sum_{i=1}^{m} |\mathcal{C}_i|$ be the average number of clusters in signature P_i. The corresponding time complexity per input $\mathbf{i}$ in COE is $O(md\log(\xi))$. In BC, we also adopt a d-dimensional tree structure over all clusters from all signatures in $\mathcal{P}$. Given $\mathbf{i}$, Pythia performs a 1NN search with $O(d\log(m\xi))$ time since it searches over all clusters from all signatures $\cup_{i=1}^{m}\mathcal{C}_i$. COE and BC require $O(md\xi)$ space. Pythia requires $O(\ell)$ and $O(1)$ communication with cohorts from $\mathcal{S}'$ and the best cohort in COE and BC schemes, respectively.

5. PYTHIA SIGNATURE UPDATE

5.1 COE signature update

Once Pythia has produced $\hat{\mathbf{x}}$ given an input, it updates $\mathcal{P}$. Only $P_i \in \mathcal{P}$, which correspond to cohorts $S_i \in \mathcal{S}''$, need to be updated. The update of P_i is based on the rule $\mathbf{c}_i^* \leftarrow \mathbf{c}_i^* + \eta_i(\mathbf{z} - \mathbf{c}_i^*)$ where $\mathbf{c}_i^* = \arg\min_{\mathbf{c} \in \mathcal{C}_i} \parallel \mathbf{z} - \mathbf{c} \parallel$, i.e., only the dimensions of $\mathbf{c}_i^*$ are modified, which correspond to the n dimensions of the non-MVs of $\mathbf{x}$. This denotes that no new clusters at P_i are formed after the update w.r.t. $\hat{\mathbf{x}}$, since $\mathbf{z} \in P_i$. The exact update can be locally reflected by $S_i \in \mathcal{S}''$ to its signature in order to be secured against a Pythia break-down situation. The magnitude of change in P_i w.r.t. $\hat{\mathbf{x}}$ is $\delta_i = \eta_i \parallel \mathbf{z} - \mathbf{c}_i^* \parallel$.

Let the sum involving the y moments of the reciprocals of binomial coefficients $F_x^{(y)} = \sum_{k=0}^{x} k^y \binom{x}{k}^{-1}$ for non-negative integers x, y. From [29] we obtain that $F_x^{(0)} = \frac{x+1}{2^{x+1}} \sum_{k=1}^{x+1} \frac{2^k}{k}$ and $F_x^{(1)} = \frac{x}{2} F_x^{(0)}$.

Theorem 4. *The expected magnitude of change in P_i, $E[\delta_i | S_i \in \mathcal{S}'']$, in COE is bounded above by $\delta_i^{\max} = \eta_i\rho_i(F_d^{(0)} - 2)$ and $\delta_i^{\max} \sim (\frac{2}{d-1} - \frac{1}{2^{d-1}})\eta_i\rho_i$ for very large d.*

Proof. Consider input $\mathbf{i}$ with $1 \leq n \leq d-1$ non-MVs and $S_i \in \mathcal{S}''$. The probability of choosing a subset of n out of d dimensions corresponding to non-MVs is $\binom{d}{n}^{-1}$. The expected magnitude of change of P_i is $E[\delta_i] = \sum_{n=1}^{d-1} \binom{d}{n}^{-1} \eta_i \parallel \mathbf{z} - \mathbf{c}_i^* \parallel \leq \sum_{n=1}^{d-1} \binom{d}{n}^{-1} \eta_i \rho_i^{(n)} \leq \sum_{n=1}^{d-1} \binom{d}{n}^{-1} \eta_i \rho_i = (F_d^{(0)} - 2)\eta_i\rho_i$. The asymptotic expansion of $F_d^{(0)} \sim 2 + \frac{2}{d-1} - \frac{1}{2^{d-1}}$ as $d \to \infty$ (proved in [30]). Hence, $\delta_i^{\max} \sim (\frac{2}{d-1} - \frac{1}{2^{d-1}})\eta_i\rho_i$. □

Theorem 5. *The expected magnitude of change in $\mathcal{P}$, $E[\delta]$, in COE is bounded above by $\delta^{\max} = \eta^{\max}\rho^{\max}(F_m^{(1)} - 1)(F_d^{(0)} - 2)$ and $\delta^{\max} \sim (m-1)(\frac{2}{d-1} - \frac{1}{2^{d-1}})\eta^{\max}\rho^{\max}$ for very large m and d, where $\eta^{\max} = \max\{\eta_i\}_{i=1}^{m}$, $\rho^{\max} = \{\rho_i\}_{i=1}^{m}$.*

Proof. The probability that a subset $\mathcal{S}''$ of ℓ cohorts is determined by Pythia is $\binom{m}{\ell}^{-1}$. Hence (from Theorem 4),

$$\begin{aligned}
E[\delta] &\leq \sum_{\ell=1}^{m} \binom{m}{\ell}^{-1} \sum_{i=1}^{\ell} \sum_{n=1}^{d-1} \binom{d}{n}^{-1} \eta_i \rho_i \\
&\leq \eta^{\max}\rho^{\max} \sum_{\ell=1}^{m} \ell \binom{m}{\ell}^{-1} (F_d^{(0)} - 2) \\
&= \eta^{\max}\rho^{\max}(F_m^{(1)} - 1)(F_d^{(0)} - 2)
\end{aligned}$$

Since $\lim_{m\to\infty} \frac{F_m^{(1)}}{m} \to 1$ (Theorem 11; [29]) and from Theorem 4, we obtain $\delta^{\max} \sim (m-1)(\frac{2}{d-1} - \frac{1}{2^{d-1}})\eta^{\max}\rho^{\max}$. □

5.2 BC signature update

The best cohort S^* updates its signature w.r.t. $\hat{\mathbf{x}}$ as described in Section 5.1, with magnitude of change bounded by $\delta^{*\max}$ (Theorem 4). Note that the change in S^*'s signature is not reflected at Pythia's $\mathcal{P}$ and specifically at the corresponding $P^* \in \mathcal{P}$.

Theorem 6. *The expected magnitude of change in $\mathcal{P}$, $E[\delta]$, in BC is bounded above by $(F_d^{(0)} - 2)\eta^{\max}\rho^{\max}$.*

Proof. Each $S_i \in \mathcal{S}$ is equally probable to be selected by Pythia as the best cohort given an input. Hence, from Theorem 5 we obtain $E[\delta] \leq \sum_{i=1}^{m} \frac{1}{m}(F_d^{(0)} - 2)\eta_i\rho_i \leq (F_d^{(0)} - 2)\eta^{\max}\rho^{\max}$. □

Pythia determines a frequency $\propto (F_d^{(0)} - 1)\eta^{\max}\rho^{\max}$ for a batch update of $\mathcal{P}$ by asking from all (previously engaged as best) cohorts to send their updated signatures changes (referring only to modified clusters), provided that they have not changed from the previous batch update. However, a batch update can be avoided once the best cohort sends the final estimate to Pythia for updating $\mathcal{P}$.

6. PERFORMANCE EVALUATION

6.1 Experiments

Setup. We conducted an extensive series of experiments to assess the performance of Godzilla, ACM and Pythia's schemes COE and BC on two real datasets (D1 and D2) and a synthetic dataset (DS). Real datasets are adopted from the UCI Machine Learning Repository [32]. D1 contains $|\mathcal{X}| = 5 \cdot 10^5$ real valued vectors of $d = 90$ corresponding to audio features. D2 contains $|\mathcal{X}| = 5 \cdot 10^4$ real valued vectors of $d = 384$ corresponding to features extracted from Computed Tomography images. Each vector of DS is a 20-dimensional point with the first fifteen dimensions randomly sampled from a Gaussian mixture of five component Gaussian pdfs with equal mixture weights and mean values of each component randomly selected from the uniform distribution $U(0, 15)$. The other five dimensions are drawn, independently, from the univariate Gaussian distribution $\mathcal{N}(0, 1)$. The first fifteen dimensions are informative dimensions, while the rest dimensions are random noises artificially added to test Pythia's capability of predicting $\mathcal{S}'$. For each dataset, we synthetically produce MVs from each $\mathbf{x}_t$ for $t = 1, \ldots, T$ as follows: each dimension $k = 1, \ldots, d$ from $\mathbf{x}_t$ is randomly and independently marked as missing with MV probability q. In this case, we expect $|\mathcal{X}| \sum_{k=1}^{d-1} \binom{d}{k} q^k (1-q)^{d-k}$ points with MVs; we exclude the cases of missing all dimensions or none. We set $q = 0.3$, which is a relatively high probability of MVs per dimension, thus, being able to test Pythia's robustness in terms of accuracy. On average, a signature P_i contains 0.32% of points of cohort's set $\mathcal{X}_i$ (this amount refers to the number of clusters stored in Pythia) using ART with initial learning rate $\eta = 0.2$, which gradually decreases. Moreover, we set the range percentage $\alpha_k = \alpha = 0.1$ for all dimensions in order to construct ρ. We run all experiments 100 times and took their average values for all performance metrics, with a stream of $T = 1000$ inputs. Pythia's schemes and MVAs

(Section 3.2) were written in Matlab. Table 1 summarizes the parameter values used in our experiments.

Parameter	Notation	Value/Range
d	dimensions	{20,90,384}
α	vigilance range pct.	0.1
η	init. learning rate	0.2
q	MV probability	0.3
m	number of cohorts	$\{5, 10, 20, 50, 100\}$
T	number of inputs	1000

Table 1: Experiment parameters.

Performance metrics. Our metrics include *efficiency metrics* and *accuracy metrics.* A scale-out system consisting of m cohorts affords two types of parallelism: *intra-imputation* and *inter-imputation* parallelism. The former refers to the capability of processing any single imputation using a number of cohorts in parallel, each accessing a dataset partition. The latter refers to the systems' capability of running in parallel a number of imputations, each of which engages a subset of cohorts. It is crucial to note that Godzilla affords neither of these parallelism types and that ACM affords only intra-imputation parallelism. This latter scenario is particularly important as typically a system is presented with a (large) batch of (vector-) inputs, each with missing values and the goal is to impute all input vectors in the batch as quickly/scalably as possible. Given this, our efficiency metrics embody various efficiency aspects impacting scalability. First, we report on *imputation latency*, defined as the time (in seconds) a system (i.e., Godzilla, ACM, Pythia-COE, or Pythia-BC) requires to impute a single input (vector) using a MVA. The rate of latency increase as dataset sizes grow is a strong aspect of scalability. In ACM, latency refers to the time a single cohort requires to impute a single input on its local dataset partition, assuming m cohorts run in parallel. In Pythia, latency refers to the time for COE / BC to predict best cohort(s) $\mathcal{S}''/S^*$, plus the latency to run MVA in parallel at cohort(s). *Imputation speedup* is defined as the ratio of Godzilla latency over ACM / COE / BC latency; it indicates how much a system is faster than Godzilla for a single imputation. *Imputation throughput* is defined as the rate of imputations delivered by a system (number of imputations per second) given a finite stream (batch) of T inputs: with this we capture the inter-imputation parallelism, *in addition to* the intra-imputation parallelism.

We measure *imputation accuracy* using the RMSE metric (root-mean squared difference) between $\mathbf{x}_a$ and $\hat{\mathbf{x}}$:

$$RMSE = \left(\frac{1}{T} \sum_{t=1}^{T} \frac{\sum_{k=1}^{d} w_{tk}(x_{(a)tk} - \hat{x}_{tk})^2}{\sum_{k=1}^{d} w_{tk}} \right)^{1/2}. \quad (4)$$

6.2 Performance results

6.2.1 Imputation efficiency

Fig. 3(a-b) shows the imputation speedup against m for all systems using KNN and REG over D2. Similar results are obtained for D1 which are omitted due to space limitations. Overall ACM, COE and BC achieve an almost linear speedup using both REG and KNN. The speedup of COE and BC drops slightly as m increases since higher m implies more signatures to be processed at Pyhtia.

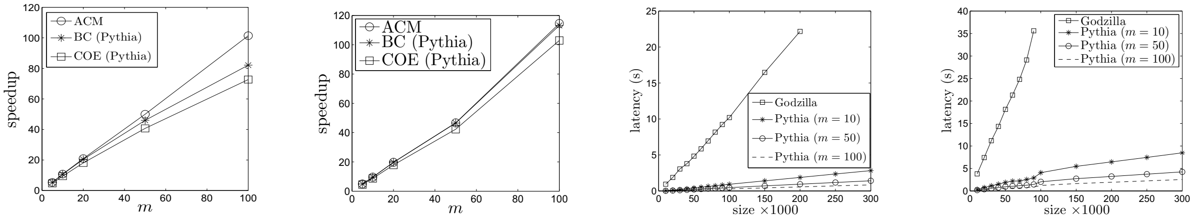
Fig. 3(c-d) shows the latency of Godzilla and Pythia-COE (Pythia-BC curves are very close to Pythia-COE) using REG when $m = \{10, 50, 100\}$ and the size of D1 and D2 varies from 5000 to 300,000 points; (similar results exist for KNN, but are omitted for space reasons). Godzilla struggles with increasing dataset sizes: with over 200,000 and 100,000 points, a high latency over 20s and 35s per input for D1 and D2, respectively, is observed. Pythia scales nicely with its latency per input increasing linearly. Moreover, when the number of cohorts increases, we obtain a sublinear increase in latency. Pythia can easily handle large datasets if more cohorts are available to scale to big data missing values.

Fig. 4(a-b) shows the throughput of each system indicating the capability of handling a stream of T inputs. COE engages S' for an input (or S^* in case of BC) thus the other cohorts ($\in S \setminus S'$) are available to be potentially engaged for other inputs in the stream. Now, recall Fig. 2(b) which shows the average number of cohorts engaged by COE per input for all data sets. For $m = 100$, about 26% of cohorts (average for all data sets) are engaged per input. Obviously, the distribution of the engaged cohorts plays an important role. That is, for a stream of inputs heading for imputation, we achieve very high throughput when (i) $|S'|$ is relatively small (in case of COE) and (ii) different imputations engage different subsets of cohorts. On the other hand, in ACM, all cohorts are concurrently occupied by the same input. The impact of the cohort engagement policy of Pythia's schemes on the throughput is illustrated in Fig. 4(a-b) using REG, where the y-axis is plotted in logarithmic scale for readability. (Similar results exist with KNN). Pythia can handle up to tens of thousands of inputs per second, compared to ACM and Godzilla, which deal with tens of inputs and a few inputs, respectively. As expected, Pythia achieves higher throughput as m increases, as the possibilities for intra-imputation parallelism increase. However, note that in Fig. 4(a) as m increases, we do not achieve further significant increase in throughput, because Pythia's processing over signatures becomes significant. The latter is higher for higher dimensions. In Fig. 4(b), as m increases, Pythia achieves high throughput. We can observe the impact of the number of dimensions d on throughput. D2 contains points with 326% more dimensions than those in D1. Pythia achieves a throughput over 10^4 (inputs/sec) with $m = 20$ in D1, while it achieves the same throughput with $m = 100$ in D2 (five times more cohorts).

Our results up to now clearly make a strong case for the scale-out advantages of the Pythia framework.

6.2.2 Imputation accuracy

Fig. 4(c-d) shows the RMSE against m using KNN and REG on synthetic data. COE and BC, as anticipated based on discussions of Example 1 and 2, obtain significant lower RMSE than Godzilla and ACM. However, this occurs with decreasing benefits as the number of cohorts increases; for $m > 50$ no further decrease in RMSE is achieved. Specifically, COE predicts a subset of cohorts, out of m cohorts, which achieves quite similar RMSE as that obtained by a subset of cohorts out of m' with $m' > m > 50$. In addition, BC engages the best cohort whose estimate is very close to the aggregate estimate of the subset of cohorts engaged by COE. Please note that ACM may yield a higher RMSE depending on the MVA used, even compared to Godzilla. For instance, using KNN, Godzilla would provide the *global* best



(a) Speedup; (KNN) (b) Speedup; (REG) (c) Latency; (REG) (d) Latency; (REG)

Figure 3: (a-b) Imputation speedup against m of ACM, COE and BC in dataset D2 using KNN and REG; (c-d) Imputation scalability against m of Godzilla and Pythia (COE) in datasets D1 & D2 using REG.

K points, whereas in ACM each cohort, even when storing irrelevant data, will be contributing its best K points. The latter necessarily implies that ACM's imputation involves points which adversely affect imputation errors.

Fig. 5 shows the RMSE against m using KNN and REG on real datasets. Pythia's schemes achieve comparable RMSE with Godzilla, with COE assuming relatively the lowest RMSE for both MVAs and datasets. In addition, the RMSE of COE remains at its lowest value from a certain m value (e.g., $m = 50$ in D2) thus there is no need to involve more cohorts. BC performs slightly better than Godzilla for both MVAs and datasets. Moreover, BC assumes higher RMSE than COE. This denotes the robustness of COE compared to BC in terms of accuracy due to the aggregate estimate from multiple engaged cohorts. ACM has higher RMSE than Godzilla in both datasets since it aggregates the estimates of all cohorts possibly incorporating estimates that spoil the final result.

6.3 Discussion

The central conclusions of our study are the following:

- Godzilla suffers from obvious severe scalability / efficiency limitations. Furthermore, it can have a poor performance even in terms of imputation accuracy.
- ACM offers efficiency performance comparable to what Map-Reduce solutions to scalability would offer, in that it requires all cohorts to be engaged for MV imputation. As such, it can only improve per-imputation efficiency. Our results show that ACM performs poorly in terms of both MV imputation throughput (compared to Pythia) and accuracy (compared to Pythia *and even* Godzilla).
- Pythia is a great all-around performer, significantly outperforming both ACM and Godzilla in terms of both overall efficiency and accuracy. Note that, even though ACM enjoys a smaller per-imputation latency than Pythia, this is achieved at a significant cost for overall imputation throughput and accuracy.
- Finally, the two Pythia schemes BC and COE, as expected can trade-off efficiency for accuracy with BC offering higher throughput but at lower accuracy.

7. CONCLUSIONS

We have tackled the problem of scaling out MV imputations, a common problem in many big data applications. We studied and developed some of the fundamentals of the problem, based on which we developed Pythia, a framework and algorithms designed for this aim. The Pythia framework is drastically different, as it on the one hand avoids the need to access all cohorts (and all associated costs for communication and for running MVAs at all cohorts), while on the other can achieve better or comparable MV imputation accuracy, compared to centralized solutions. Specifically, our comprehensive experiments showed that it can provide drastically better efficiency/scalability and accuracy compared to a centralized approach (Godzilla) and a massively parallel, a la Map-Reduce, solution (ACM). Future work plans entail the study of additional cohort prediction schemes, straddling the line between efficiency and accuracy.

8. ACKNOWLEDGEMENTS

This research has been co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program 'Education and Lifelong Learning' of the National Strategic Reference Framework (NSRF) - Research Funding Program: Thales. Investing in knowledge society through the European Social Fund.

9. REFERENCES

[1] X. Su, *et al*, 'Using Classifier-Based Nominal Imputation to Improve Machine Learning', *Proc. 15th PAKDD*, Part I, LNAI 6634, pp. 124–135, 2011.

[2] A. Farhangfar, *et al*, 'Impact of imputation of missing values on classification error for discrete data', *Pattern Recognition*, 41(12): 3692–3705, Dec 2008.

[3] M.T. Asif, *et al*, 'Low–Dimensional Models for Missing Data Imputation in Road Networks', *Proc. 38th IEEE ICASSP*, pp.3527–3531, 2013.

[4] E.C. Chi, *et al*, 'Genotype imputation via matrix completion', *Genome Research*, 23(3):509–18, Mar 2013.

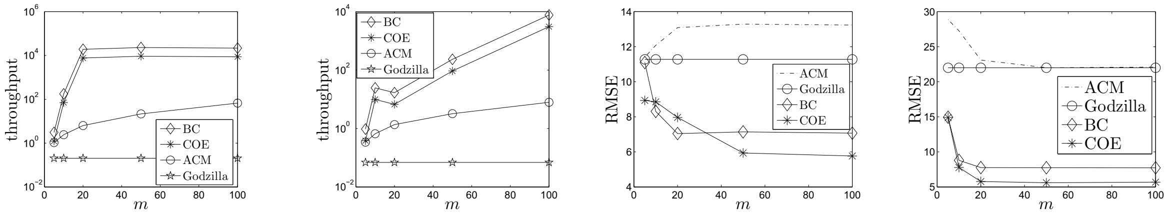
[5] I.B. Aydilek, *et al*, 'A novel hybrid appoach to estimating missing values in databases using k–nearest neighbors and neural networks', *Innovative Computing, Information and Control*, 8(7A): 1349–4198, Jul 2012.

[6] A. Farhangfar, *et al*, 'A Novel Framework for Imputation of Missing Values in Databases', *IEEE Trans. Sys. Man Cyber. (A)*, 37(5): 692–709, Sep 2007.

[7] K. Lakshminarayan, *et al*, 'Imputation of missing data in industrial databases', *Appl. Intell.*, 11(3): 259–275, Nov / Dec 1999.

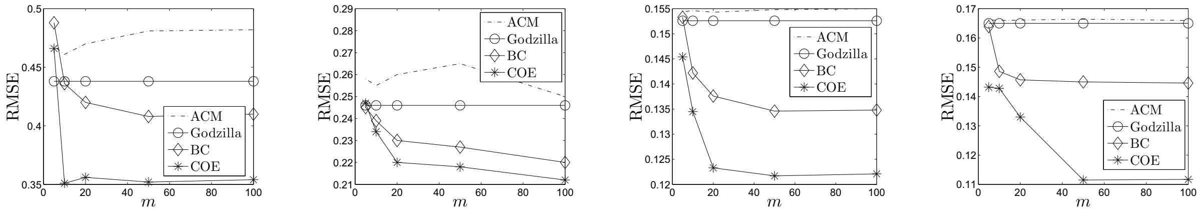
[8] L. A. Kurgan, *et al*, 'Mining the cystic fibrosis data', J. Zurada & M. Kantardzic (Eds.), *Next Generation of Data–Mining Applications*, IEEE Press, 415–444, 2005.

[9] A.W. Liew, *et al*, 'Missing value imputation for gene expression data: computational techniques to recover missing data from available information', *Brief. Bioinform.*, 12(5): 498–513, Sep 2011.



(a) Throughput; REG (b) Throughput; REG (c) RMSE; KNN (d) RMSE; REG

Figure 4: (a-b) System throughput against m of Godzilla, ACM, COE and BC in dataset D1 & D2 using REG; (c-d) RMSE against m of Godzilla, ACM, COE and BC in dataset DS using KNN and REG.



(a) RMSE; D1 (KNN) (b) RMSE; D1 (REG) (c) RMSE; D2 (KNN) (d) RMSE; D2 (REG)

Figure 5: RMSE against m of Godzilla, ACM, COE and BC in dataset D1 (a-b) and D2 (c-d) using KNN and REG.

[10] J. Dean, *et al*, 'MapReduce: Simplified Data Processing on Large Clusters', *Proc. USENIX OSDI*, 2004.

[11] S. Ghemawat, *et al*, 'The Google File System', *Proc. ACM SOSP*, 2003.

[12] C-T. Chu, *et al*, 'Map-Reduce for Machine Learning on Multicore', *NIPS 19*, MIT press, 281–288, 2006.

[13] C. K. Enders, 'Applied Missing Data Analysis', *Guilford Press*, NY, 2010.

[14] D. W. Joenssen, *et al*, 'Hot Deck Methods for Imputing Missing Data', *Proc. 8th MLDM* , LNCS 7376, pp.63–75, 2012.

[15] O. Troyanskaya, *et al*, 'Missing value estimation methods for DNA microarrays', *Bioinformatics*, 17(6):520–525, 2001.

[16] R.J. Little, *et al*, 'Statistical Analysis with Missing Data', *Wiley*, NY, 1987.

[17] T.E. Raghunathan, *et al*, 'A multivariate technique for multiply imputing missing values using a sequence of regression models', *Survey Methodology*, 27(1):85–95, 2001.

[18] D.B. Rubin, 'Multiple Imputation After 18+ Years', *J. of the American Statistical Association*, 91(434):473–489, 1996.

[19] L. Li, *et al*, 'DynaMMo: mining and summarization of coevolving sequences with missing values', *Proc. 15th KDD*, 527–534, 2009.

[20] S. Yang, *et al*, 'Online recovery of missing values in vital signs data streams using low–rank matrix completion', *Proc. 11th IEEE ICMLA*, 281–287, 2012.

[21] M. Ouyang, *et al*, 'Gaussian mixture clustering and imputation of microarray data', *Bioinformatics*, 20(6): 917–923, Apr 2004.

[22] T. Aittokallio, *et al*, 'Dealing with missing values in large-scale studies: microarray data imputation and beyond' *Brief. Bioinform.* 11(2):253–264, 2010.

[23] D-W. Kim, *et al*, 'Iterative Clustering Analysis for Grouping Missing Data in Gene Expression Profiles', *Proc. PAKDD 2006*, LNAI 3918, pp.129–138, 2006.

[24] M.R. Garey, *et al*, 'Computers and Intractability; A Guide to the Theory of NP–Completeness', *W. H. Freeman & Co.*, NY, 1990.

[25] B. Przydatek, 'A fast approximation algorithm for the subset–sum problem', *Intl. Trans. in Op. Res.*, 9(4): 437–459, Jul 2002.

[26] G. A. Carpenter, *et al*, 'The ART of adaptive pattern recognition by a self–organizing neural network', *IEEE Computer*, 21(3): 77–88, Mar 1988.

[27] A. Ahmad, *et al*, 'A k–mean clustering algorithm for mixed numeric and categorical data' *Data & Knowledge Engineering*, 63(2):503–527, 2007.

[28] P. J. Rousseeuw, *et al*, 'Alternatives to the median absolute deviation', *J. American Statistical Association*, 88(424): 1273–1283, Dec 1993.

[29] H. Belbachir, *et al*, 'Sums involving moments of reciprocals of binomial coefficients', *J. Integer Sequences*, 14(6), Article 11.6.6, 16p, 2011.

[30] J-H. Yang, *et al*, 'The asymptotic expansions of certain sums involving inverse of binomial coefficient', *Intl. Mathematical Forum*, 5(16): 761–768, 2010.

[31] J.L. Bentley, 'Multidimensional binary search trees used for associative searching', *Communications of the ACM*, 18(9):509–517, 1975.

[32] K. Bache, *et al*, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml] Irvine, Uni. of California, School of Inform. and Comp. Sci., 2013.