Machine Learning (2021) 110:621–650
https://doi.org/10.1007/s10994-021-05954-3

An extended DEIM algorithm for subset selection and class identification

Emily P. Hendryx[1] · **Béatrice M. Rivière**[2] · **Craig G. Rusin**[3]

Received: 17 January 2020 / Revised: 7 August 2020 / Accepted: 2 February 2021 /
Published online: 21 March 2021

Abstract
The discrete empirical interpolation method (DEIM) has been shown to be a viable index-selection technique for identifying representative subsets in data. Having gained some popularity in reducing dimensionality of physical models involving differential equations, its use in subset-/pattern-identification tasks is not yet broadly known within the machine learning community. While it has much to offer as is, the DEIM algorithm is limited in that the number of selected indices cannot exceed the rank of the corresponding data matrix. Although this is not an issue for many data sets, there are cases in which the number of classes represented in a given data set is greater than the rank of the data matrix; in such cases, it is impossible for the standard DEIM algorithm to identify all classes. To overcome this issue, we present a novel extension of DEIM, called E-DEIM. With the proposed algorithm, we also provide some theoretical results for using extensions of DEIM to form the CUR matrix factorization in identifying both rows and columns to approximate the original data matrix. Results from applying variations of E-DEIM to two different data sets indicate that the presented extension can indeed allow for the identification of additional classes along with those selected by standard DEIM. In addition, comparing these results to those of some more familiar methods demonstrates that the proposed deterministic E-DEIM approach including coherence performs comparably to or better than the other evaluated methods and should be considered in future class-identification tasks.

Keywords Subset selection · Class identification · Discrete empirical interpolation method · Low rank data

1 Introduction

Dimension reduction techniques often play an important role in the analysis of large data sets. There are a number of different dimension reduction techniques that can be applied to a given problem, but often these reduced data sets consist of derived features that are no

Editor: Jean-Philippe Vert.

✉ Emily P. Hendryx
ehendryx@uco.edu

Extended author information available on the last page of the article

longer interpretable in their original context. For instance, in using a method such as principal components analysis (PCA) on medical data, it would likely be difficult for physicians to apply their expert training to interpret the clinical meaning of each individual principal component at face value. Hence, in some settings, it is necessary to identify a dimension reduction technique that preserves the original structure of the data so as not to lose its interpretability, selecting a meaningful subset of observations or features prior to conducting further analysis. This task of identifying class/group representatives in an unsupervised manner can prove challenging as there may be little, or no, prior knowledge about the number of classes present in data set.

As demonstrated in a recent paper by Hendryx et al. (2018), one such way to reduce the data dimension while preserving data interpretability is to use the discrete empirical interpolation method (DEIM) index-selection algorithm. Presented as part of a CUR matrix factorization for identifying important rows and columns of a given data matrix, DEIM index selection is the underlying means for selecting said rows and columns (Hendryx et al., 2018). This index-selection method makes use of the underlying linear-algebraic structure of the data to determine some of the most influential rows and columns defining the space in which the data lives.

While previous works demonstrate the utility of DEIM, the algorithm does have its limitations. In this work, we address the scenario in which DEIM is unable to identify all of the classes in the data merely due to the rank of the corresponding data matrix of interest; the number of DEIM-selected rows or columns inherently cannot exceed the matrix rank. In practice, however, there are some cases in which the number of classes in the data is expected to exceed the rank of the matrix. For instance, if the number of derived features or samples in time, in the case of time series observations is small for a data set representing a greater number of classes, DEIM can at best only detect as many classes as the number of features/time samples per observation. As data sets are getting larger and larger, the likelihood that the number of observations vastly exceeds the number of features/samples per observation is increasing. One such example of particular interest to the authors [see the dissertation by Hendryx (2018)] is class identification in the scenario in which hundreds of millions, if not billions, of ECG tracings of cardiac cycles have been recorded in a clinical setting across a diverse patient population; where the length of each beat observation may be limited to a few hundred samples per beat, it is not unreasonable to expect the presence of a far greater number of distinct beat morphologies. Another specific example of a case in which the number of classes is expected to exceed the data matrix rank is presented in the Letter Recognition Data Set (Frey & Slate, 1991; Dua & Taniskidou, 2017), described and analyzed further below. In this work, we present extensions of DEIM designed to accommodate subset selection in such data sets. Subset selection/class identification in this type of data can play an important role in providing a summary of the data, either for real-time interpretation or for the further development of predictive models.

We also note that since DEIM relies on an approximation to the singular value decomposition (SVD) of a matrix, the proposed approach may also find use in the setting in which a rank-k SVD approximation is prohibitively expensive to compute even for moderately-small k. In this setting, our extension of DEIM allows for the selection of additional indices without the need to compute the full rank-k approximation, leveraging a lower-rank approximation instead. Hence, this notion of extending DEIM can be applied in a number of contexts. With index oversampling having been explored more in the model reduction community [for example, see works by Zhou (2012), Zimmermann and Willcox (2016), and Peherstorfer et al. (2018), to name a few], future studies comparing the proposed extensions of DEIM via oversampling in both the machine learning and model

reduction applications are necessary. For the scope of this work, however, we primarily focus on our method's performance in class identification. We also briefly discuss theoretical implications of using such an extension in computing the CUR matrix factorization to approximate the original data matrix, with further theoretical studies suggested as a topic for future work.

Following a brief review of the standard DEIM implementation presented by Chaturantabut and Sorensen (2010) and Sorensen and Embree (2016), we describe our proposed DEIM extension. After further discussion regarding the reasoning behind the algorithm design, we then present some theoretical results that follow from extending DEIM to the CUR factorization context. Finally, we provide results from comparing extended DEIM to two other more commonly known selection methods applied to two different data sets, demonstrating the effectiveness of extended DEIM methods in detecting additional classes.

Notation In the sections that follow, for a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, we use $\mathbf{a}_j = \mathbf{A}(:, j)$ to indicate the jth column of $\mathbf{A}$ and $\mathbf{A}_\mathbf{t} = \mathbf{A}(:, \mathbf{t})$ to represent the columns of $\mathbf{A}$ corresponding to the indices held in the vector $\mathbf{t} \in \mathbb{N}^\ell$ for $1 \leq \ell \leq n$.

The algorithms in this work are presented using MATLAB notation. Of particular note is that "$[\cdot, \cdot] = \max(\cdot)$" takes as input a vector and outputs the maximum entry of the vector followed by the index of said maximal entry. In addition, the operation "$.*$" performs element-wise vector multiplication.

2 Background

The discrete empirical interpolation method (DEIM) was initially introduced by Chaturantabut and Sorensen (2010) within the context of performing model reduction. In particular, the method is presented with the goal of reducing the order of systems of ordinary differential equations containing nonlinearities. DEIM is extended to the formation of the CUR matrix factorization in a recent work by Sorensen and Embree (2016). Hendryx, Rivière, Sorensen, and Rusin apply this DEIM-CUR matrix factorization to the medical domain for the identification of representative electrocardiogram (ECG) beat morphologies (2018); the DEIM-selected beats are shown to be representative of the larger data set, providing a means of summarizing the data for additional analyses. In particular, the beats selected in this manner can be used in the classification of the remaining, unselected beats in the data set for the development of clinical decision support tools.

Given the promising results seen in previous work, we build off of the DIEM algorithm to select additional indices. Before presenting our extension of DEIM, however, we first turn to a description of the construction and implementation of the original, "standard" DEIM algorithm.

2.1 Standard DEIM

The DEIM algorithm provides a means of approximating an $m \times n$ matrix $\mathbf{A}$ in a space of dimension $k \leq \text{rank}(\mathbf{A})$ in such a way that k of the original matrix rows are preserved exactly. In particular, this k-dimensional space is chosen to be that spanned by the left singular vectors resulting from the rank-k SVD approximation to $\mathbf{A}$. The rank-k SVD of $\mathbf{A}$ is given by $\mathbf{A} \approx \mathbf{V}\mathbf{S}\mathbf{W}^T$, where $\mathbf{V} \in \mathbb{R}^{m\times k}$ and $\mathbf{W} \in \mathbb{R}^{n\times k}$ have orthonormal columns, and $\mathbf{S}$ is a diagonal matrix containing the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k > 0$. It is well understood that the SVD yields the optimal rank-k approximation to $\mathbf{A}$ with respect to the induced matrix 2-norm.

For the purposes of DEIM, then, the columns of the matrix $\mathbf{V}$—that is $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$—form a basis for the space in which we approximate the matrix $\mathbf{A}$. Then,

$$\mathbf{A} \approx \mathbf{VB},$$

where $\mathbf{B} \in \mathbb{R}^{k \times n}$ is the coefficient matrix to be defined such that this approximation preserves k rows of $\mathbf{A}$ exactly. Suppose the index vector $\mathbf{p} \in \mathbb{N}^k$ is such that it contains non-repeating row indices p_1, p_2, ..., p_k corresponding to the rows to be preserved with $1 \leq p_j \leq m$ for $1 \leq j \leq k$. Then we set $\mathbf{P} = [\mathbf{e}_{p_1}, \mathbf{e}_{p_2}, \ldots, \mathbf{e}_{p_k}] = \mathbf{I}(:, \mathbf{p})$, where $\mathbf{e}_j$ is defined as the vector of length m containing all zeros except for a 1 in the jth entry (or the jth column of the $m \times m$ identity matrix, $\mathbf{I}$, denoted as $\mathbf{I}(:, p_j)$). To maintain the rows of interest in the matrix approximation, we require that

$$\mathbf{P}^T\mathbf{A} = \mathbf{P}^T(\mathbf{VB}).$$

Then for invertible $\mathbf{P}^T\mathbf{V}$, solving for the coefficient matrix $\mathbf{B}$ yields

$$\mathbf{B} = (\mathbf{P}^T\mathbf{V})^{-1}\mathbf{P}^T\mathbf{A}.$$

Hence, we arrive at the approximation

$$\mathbf{A} \approx \mathbf{V}(\mathbf{P}^T\mathbf{V})^{-1}\mathbf{P}^T\mathbf{A} = \mathcal{P}\mathbf{A},$$

where

$$\mathcal{P} = \mathbf{V}(\mathbf{P}^T\mathbf{V})^{-1}\mathbf{P}^T$$

is DEIM's interpolatory projector. (Note that if we want to preserve columns of $\mathbf{A}$ instead of rows, we can use a similar process to form a different interpolatory projector using the right singular vectors held in $\mathbf{W}$.)

With the approximation space selected to be the span of the left singular vectors, the remaining question, then, is how to select the indices in forming $\mathbf{P}$. While subsequent works have proposed alternative approaches to selecting these indices [for example, see the QDEIM algorithm proposed by Drmač and Gugercin (2016)], we focus only on the original DEIM algorithm proposed by Chaturantabut and Sorensen (2010). The extension of other such index-selection procedures for class-identification purposes is not included here, but it is a topic of interest for future studies.

2.1.1 Construction of $\mathbf{p}$ in standard DEIM

To select the indices held in $\mathbf{p}$, each column of $\mathbf{V}$ is considered in turn, with $\mathbf{p}(1) = \arg\max_i |\mathbf{V}(i, 1)|$. The subsequent index in $\mathbf{p}(j+1)$, denoted as p_{j+1}, is determined by subtracting from $\mathbf{v}_{j+1}$ the interpolatory projection of $\mathbf{v}_{j+1}$ onto the range of $\mathbf{V}_j = \mathbf{V}(:, 1:j)$, defining $\mathbf{p}(j+1)$ to be the index corresponding to the largest element of this difference in absolute value. That is, for $\mathbf{r} = \mathbf{v}_{j+1} - \mathcal{P}_j\mathbf{v}_{j+1}$, p_{j+1} is selected such that $\mathbf{r}(p_{j+1}) = \|\mathbf{r}\|_\infty$, where $\mathcal{P}_j = \mathbf{V}_j(\mathbf{P}_j\mathbf{V}_j)^{-1}\mathbf{P}_j^T$ and $\mathbf{P}_j = \mathbf{I}(:, \mathbf{p}_j)$ for $\mathbf{p}_j$ containing the first j indices in $\mathbf{p}$. Notice that the interpolatory nature of $\mathcal{P}_j$ ensures that $\mathbf{r}(p_i) = 0$ for all $i < (j+1)$ since

$$\mathbf{r}(\mathbf{p}_j) = \mathbf{P}_j^T\mathbf{r} = \mathbf{P}_j^T(\mathbf{v}_{j+1} - \mathcal{P}_j\mathbf{v}_{j+1}) = \mathbf{P}_j^T\mathbf{v}_{j+1} - \mathbf{P}_j^T\mathbf{V}_j(\mathbf{P}_j^T\mathbf{V}_j)^{-1}\mathbf{P}_j^T\mathbf{v}_{j+1} = \mathbf{0}.$$

Hence, the indices selected by DEIM are guaranteed to be unique. In addition, this approach to selecting the k indices in $\mathbf{p}$ ensures that that $\mathbf{P}^T\mathbf{V}$ is indeed invertible for DEIM;

a proof of this is provided in Lemma 3.2 from Sorensen's and Embree's (2016) paper. The full procedure for selecting the DEIM indices given $\mathbf{V} \in \mathbb{R}^{m \times k}$ is outlined in more precise detail in Algorithm 1.

Algorithm 1 DEIM Point Selection (Adapted from work by Sorensen and Embree (2016))

Input: $\mathbf{V}$, a matrix in $\mathbb{R}^{m \times k}$ with $m > k$

Output: $\mathbf{p}$, a vector in $\mathbb{N}^k$ containing integral values from $\{1, .., m\}$

$\mathbf{v} = \mathbf{v}_1$
$[\sim, p_1] = \max(|\mathbf{v}|)$
$\mathbf{p} = p_1$
for $j = 2 : k$ **do**
  $\mathbf{v} = \mathbf{v}_j$
  $\mathbf{c} = \mathbf{V}(\mathbf{p}, 1 : j-1)^{-1}\mathbf{v}(\mathbf{p})$
  $\mathbf{r} = \mathbf{v} - \mathbf{V}(:, 1 : j-1)\mathbf{c}$
  $[\sim, p_j] = \max(|\mathbf{r}|)$
  $\mathbf{p} = [\mathbf{p}; p_j]$
end for

In the model reduction context, the DEIM indices selected in Algorithm 1 were originally selected to "limit growth of an error bound"—specifically the approximation error determined by $\|\mathbf{x} - \mathcal{P}\mathbf{x}\|_2$ for an arbitrary vector $\mathbf{x} \in \mathbb{R}^m$ approximated in the k-dimensional space spanned by the set of orthonormal vectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ with $k \leq m$ [see Lemma 3.2 by Chaturantabut and Sorensen (2010)]. While this type of error bound is important in many contexts, the role of such a bound in class identification is not necessarily clear at this point. However, we make two observations about this algorithm in application to class-identification tasks: (1) the use of the left singular vectors in forming the approximation space is very closely related to the popular unsupervised learning method, PCA, leveraging the properties afforded by an optimal lower-rank approximation to the space in which the data lives, and (2) the selection of indices from the residual vector, $\mathbf{r}$, essentially boils down to identifying rows that contribute the most new information in the construction of the interpolatory projector, a projector constructed specifically to preserve particular rows. It is this intuition regarding the workings of the standard DEIM approach that we hold on to and build off of in extending this algorithm to select additional indices.

3 Proposed method: extended DEIM (E-DEIM)

Suppose, now, that we want to select $\hat{k}$ separate rows from $\mathbf{A} \in \mathbb{R}^{m \times n}$, using the columns of the full rank matrix $\mathbf{V} \in \mathbb{R}^{m \times k}$ to span the approximation space, and construct $\mathbf{p} \in \mathbb{N}^{\hat{k}}$ via a DEIM-type approach, where $k \leq \hat{k} \leq m$. Then letting $\mathbf{P} = \mathbf{I}(:, \mathbf{p})$ in $\mathbb{R}^{m \times \hat{k}}$, $\mathbf{P}^T\mathbf{V}$ in $\mathbb{R}^{\hat{k} \times k}$ is not invertible (given $\hat{k} \neq k$), and we use a pseudoinverse of $\mathbf{P}^T\mathbf{V}$ to define $\mathcal{P}$ as

$$\mathcal{P} = \mathbf{V}(\mathbf{P}^T\mathbf{V})^{\dagger}\mathbf{P}^T, \tag{1}$$

where here we use the left Moore–Penrose pseudoinverse

$$(\mathbf{P}^T\mathbf{V})^{\dagger} = [(\mathbf{P}^T\mathbf{V})^T(\mathbf{P}^T\mathbf{V})]^{-1}(\mathbf{P}^T\mathbf{V})^T,$$

assuming $\mathbf{P}^T\mathbf{V}$ has full column rank.[1]

Note that

$$\begin{aligned}\mathcal{P}^2 &= \mathbf{V}(\mathbf{P}^T\mathbf{V})^\dagger\mathbf{P}^T\mathbf{V}(\mathbf{P}^T\mathbf{V})^\dagger\mathbf{P}\\ &= \mathbf{V}(\mathbf{P}^T\mathbf{V})^\dagger\mathbf{P}^T\\ &= \mathcal{P},\end{aligned}$$

indicating that $\mathcal{P}$ is indeed still a projection.

Also notice that by increasing the length of $\mathbf{p}$ and choosing the left inverse of $\mathbf{P}^T\mathbf{V}$ to form the projector, we lose the interpolatory nature for general $\mathbf{z} \in \mathbb{R}^m$ while maintaining the interpolatory nature of $\mathcal{P}$ for vectors in the range of $\mathbf{V}, \mathcal{R}(\mathbf{V})$. Suppose, for example, $\mathbf{y} \in \mathcal{R}(\mathbf{V})$. Then there exists $\mathbf{x} \in \mathbb{R}^k$ such that $\mathbf{y} = \mathbf{V}\mathbf{x}$ and

$$\begin{aligned}(\mathcal{P}\mathbf{y})(\mathbf{p}) &= \mathbf{P}^T\mathcal{P}\mathbf{V}\mathbf{x} = \mathbf{P}^T\mathbf{V}(\mathbf{P}^T\mathbf{V})^\dagger\mathbf{P}^T\mathbf{V}\mathbf{x}\\ &= \mathbf{P}^T\mathbf{V}\mathbf{x} = \mathbf{y}(\mathbf{p}).\end{aligned}$$

3.1 Construction of p for extended DEIM

A question for the case in which $\hat{k} > k$ is: How should $\mathbf{p}(k+1 : \hat{k})$ be selected? For example, one could simply select a random subset of an additional $\hat{k} - k$ indices beyond those selected by standard DEIM, or the additional indices could be selected by looking at the next largest residuals held in $\mathbf{r}$ in the final iteration of the for loop in Algorithm 1. In the model reduction literature, some ideas have included identifying indices that reduce the projection error through additional SVD computations (Zimmermann and Willcox, 2016; Peherstorfer et al., 2018) and leveraging nonlinear variables from the model of interest (Zhou, 2012). In a recent work, Manohar et al. (2018) use a pivoted QR factorization to identify additional indices in the setting of sensor placement selection in control theory, an application task similar to the machine learning tasks of subset/feature selection discussed herein. In our approach, the starting objective is simply to select $\mathbf{p}(k+1 : \hat{k})$ such that $\mathbf{A}(\mathbf{p}(k+1 : \hat{k}), :)$ is guaranteed to have full row rank (requiring that $\hat{k} \leq 2k$ for a single application of a DEIM variant); then the selected sub-matrix $\mathbf{A}(\mathbf{p}, :)$ will contain two blocks of linearly independent rows, and we can make further use of some of the linear-algebraic properties of the data if needed.

To select a second set of $\hat{k} - k$ linearly independent rows of $\mathbf{A}$, our approach performs standard DEIM to select the first k indices and then applies a modified a version of DEIM—referred to as "restarted DEIM" below—to a submatrix of $\mathbf{V}$ to find the additional $\hat{k} - k$ indices. This submatrix, $\hat{\mathbf{V}}$, is initially taken to contain all of the rows of $\mathbf{V}$ not in $\mathbf{V}(\mathbf{p}, :)$; that is, $\hat{\mathbf{V}} = \mathbf{V}(\mathbf{p}^c, :)$, where $\mathbf{p}^c \in \mathbb{N}^{m-k}$ contains those indices from 1 to m not contained in $\mathbf{p}$.

In selecting the additional indices using $\hat{\mathbf{V}}$, we consider two residuals, $\mathbf{r}_1$ and $\mathbf{r}_2$, in restarted DEIM. The formation of these residuals is discussed below, with variations on the use of $\mathbf{r}_1$ and $\mathbf{r}_2$ studied in the results section.

[1] Given that the extended DEIM approach presented here first makes use of the standard-DEIM-selected indices, this assumption on the rank of $\mathbf{P}^T\mathbf{V}$ holds for the work herein.

3.1.1 Forming $\mathbf{r}_1$

Similar to the formation of $\mathbf{r}$ in standard DEIM, $\mathbf{r}_1$ contains the residual from a complementary projection computed in constructing $\mathbf{p}$. However, since removing rows of $\mathbf{V}$ to form $\hat{\mathbf{V}}$ means that $\hat{\mathbf{V}}$ is no longer guaranteed to have full column rank, at the ith iteration of restarted DEIM, the residual produced by $(\mathbf{I} - \mathcal{P}_{j-1})\hat{\mathbf{v}}_i$ is checked, where $\mathcal{P}_{j-1}$ is the projector to be described further below. If the magnitude of the largest residual entry is too small, then $\hat{\mathbf{v}}_i$ lies too close to the range of the previously considered columns of $\hat{\mathbf{V}}$ and should be removed from future consideration.

3.1.2 Forming $\mathbf{r}_2$

If a given row of $\hat{\mathbf{V}}$ is too similar to the rows of $\mathbf{V}(\mathbf{p}, :)$, then that row should not be selected for inclusion in $\mathbf{p}$. The role of the $\mathbf{r}_2$ residual, then, is to help maintain a form of "memory" of the previously selected rows from the first (standard) DEIM implementation. However, this requires that we identify a notion of "too similar" for this context. We consider a few different ideas here. Note that unlike $\mathbf{r}_1$, in each of the approaches presented, the residual in $\mathbf{r}_2$ is computed only prior to restarting DEIM.

One way that we compare $\hat{\mathbf{V}}$ and $\mathbf{V}(\mathbf{p}, :)$ is to compute the ℓ_1 distance between each row of $\hat{\mathbf{V}}$ and the rows of $\mathbf{V}(\mathbf{p}, :)$, defining $\mathbf{r}_2$ such that

$$\mathbf{r}_2(i) = \min_{j \in \mathbf{p}} \|\mathbf{V}(j, :) - \hat{\mathbf{V}}(i, :)\|_1.$$

Depending on the data set, we may want to replace the ℓ_1 distance with another measure of distance, for example dynamic time warping. This residual vector is then normalized by its maximum entry so that it contains values between 0 and 1 with values closer to 0 corresponding to rows that are most similar to those already selected in standard DEIM.

We can also compare $\hat{\mathbf{V}}$ and $\mathbf{V}(\mathbf{p}, :)$ by looking at the angles between their rows. This idea stems from the notion of the *coherence* (or *mutual coherence*) of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ defined in the information theoretic and signal recovery literature—see, for example, works by Donoho and Huo (2001) and Candes et al. (2011)—as

$$\mu(\mathbf{A}) = \max_{i \neq j} \frac{|\mathbf{a}_i^T \mathbf{a}_j|}{\|\mathbf{a}_i\|_2 \|\mathbf{a}_j\|_2}.$$

Notice that $\mu(\mathbf{A})$ is bounded between 0 and 1 and simply corresponds to the cosine of the smallest angle between the columns of $\mathbf{A}$; smaller values of $\mu(\mathbf{A})$ indicate that the columns of $\mathbf{A}$ are relatively spread out in $\mathbb{R}^m$, where a larger coherence indicates that there are at least two columns in $\mathbf{A}$ that lie close to each other. In the signal recovery context, there is often a need for the data observations to be "spread out", or incoherent, in order to recover the "true" data. Here, we want to encourage selection of additional rows from $\hat{\mathbf{V}}$ that are "spread out" from those already selected in $\mathbf{V}(\mathbf{p}, :)$.

Using this notion, we normalize the rows of $\hat{\mathbf{V}}$ and $\mathbf{V}(\mathbf{p}, :)$ via the ℓ_2-norm to get $\hat{\mathcal{V}}$ and $\mathcal{V}(\mathbf{p}, :)$, respectively, and define $\mathbf{r}_2$ as

$$\mathbf{r}_2(i) = 1 - \max_j(|\hat{\mathcal{V}}(i, :)\mathcal{V}(p_j, :)^T|),$$

where the maximum is taken in a row-wise fashion. The subtraction of the second term from 1 forces larger entries in $\mathbf{r}_2$ to correspond to those rows of $\hat{\mathbf{V}}$ that are most different

from the previous DEIM-selected rows. The need for this will become more apparent as we discuss combining the information held in $\mathbf{r}_1$ and $\mathbf{r}_2$ below.

3.1.3 Combining $\mathbf{r}_1$ and $\mathbf{r}_2$

To take into account the indices selected in the standard DEIM step as well as the proximity of $\hat{\mathbf{v}}_i$ to the previously considered columns in restarted DEIM, the different information carried by $\mathbf{r}_1$ and $\mathbf{r}_2$ is combined into a single vector, taking the element-wise product of the two vectors:

$$\hat{\mathbf{r}} = \mathbf{r}_1 . * \mathbf{r}_2. \tag{2}$$

This particular means of combining the two residuals is chosen because it allows each residual to simultaneously influence the selection of an index while still guaranteeing that the restarted DEIM projection is well-defined (discussed more rigorously below). While the inclusion of $\mathbf{r}_2$ can be considered optional, as suggested by the experiments presented later in this work, the information in $\mathbf{r}_1$ is critical for theoretical guarantees. Hence, rather than forming the composite residual $\hat{\mathbf{r}}$ by adding $\mathbf{r}_1$ and $\mathbf{r}_2$, we scale $\mathbf{r_1}$ by the entries in $\mathbf{r}_2$, where $\mathbf{r}_2$ is specifically constructed such that $0 \leq \mathbf{r}_2(i) \leq 1$ for $1 \leq i \leq (m-k)$ to ensure the magnitude of $\hat{\mathbf{r}}$ cannot exceed that of $\mathbf{r}_1$.

With that in mind, if the maximum entry of $\hat{\mathbf{r}}$ in absolute value is below a user-defined tolerance, τ, then it is likely that no new information is contributed by $\hat{\mathbf{v}}_i$ for index selection. Otherwise, this residual is used to select the next index to be included in $\hat{\mathbf{p}}$—the vector of indices formed from $\hat{\mathbf{V}}$—and $\hat{\mathbf{v}}_i$ is included in forming the projector for the next iteration.

In initializing restarted DEIM, $\hat{\mathbf{r}}$ is computed as $\hat{\mathbf{r}} = \hat{\mathbf{v}}_1 . * \mathbf{r}_2$, and the first entry in $\hat{\mathbf{p}}$ is determined by the maximum entry (in absolute value) of $\hat{\mathbf{r}}$ if it exceeds the given tolerance, τ; if $\|\hat{\mathbf{r}}\|_\infty$ is less than τ, then $\hat{\mathbf{r}}$ is recomputed using the next column of $\hat{\mathbf{V}}$, repeating the process until the tolerance is surpassed. If the tolerance is never exceeded, then the algorithm terminates having only performed standard DEIM. Otherwise, the algorithm then proceeds to find the rest of the restarted DEIM indices starting with the next column of $\hat{\mathbf{V}}$. With this approach, the projector at iteration $j+1$ is given by

$$\hat{\mathcal{P}}_j = \hat{\mathbf{V}}_{\mathbf{t}_j} (\hat{\mathbf{P}}_j^T \hat{\mathbf{V}}_{\mathbf{t}_j})^{-1} \hat{\mathbf{P}}_j^T,$$

where $\hat{\mathbf{P}}_j = \mathbf{I}(:, \hat{\mathbf{p}}_j)$ for $\hat{\mathbf{p}}_j = \hat{\mathbf{p}}(1 : j)$, and $\mathbf{t}_j$ contains the indices of those columns of $\hat{\mathbf{V}}$ that have met the residual tolerance criterion in the previous iterations. Notice that by carefully constructing $\mathbf{t}_j$ through our treatment of $\hat{\mathbf{r}}$, we have ensured that $(\hat{\mathbf{P}}_j^T \hat{\mathbf{V}}_{\mathbf{t}_j})^{-1}$ exists at each iteration. This claim is restated below in Lemma 1; the presented proof of this lemma uses induction and closely follows the structure of the proof of a similar claim in Lemma 3.2 from the work by Sorensen and Embree (2016).

Lemma 1 *Suppose* $\hat{\mathbf{V}} = [\hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_k]$ *in* $\mathbb{R}^{m \times k}$ *has rank* γ.

Let $\hat{\mathbf{P}}_j = [\mathbf{e}_{\hat{p}_1}, \mathbf{e}_{\hat{p}_2}, \ldots, \mathbf{e}_{\hat{p}_j}]$ *and let* $\hat{\mathbf{V}}_{\mathbf{t}_j} = \hat{\mathbf{V}}(:, \mathbf{t}_j)$ *for* $1 \leq j \leq \gamma$, *where* $\mathbf{t}_j \in \mathbb{N}^j$ *contains non-repeating values in* $\{1, \ldots, k\}$. *If the entries in* $\mathbf{t}_j$ *are selected via the proposed DEIM extension, then* $\hat{\mathbf{P}}_j^T \hat{\mathbf{V}}_{\mathbf{t}_j}$ *is nonsingular for* $1 \leq j \leq \gamma$.

Proof First, note that $\hat{\mathbf{P}}_1^T \hat{\mathbf{V}}_{\mathbf{t}_1} = \mathbf{e}_{\hat{p}_1}^T \hat{\mathbf{V}}_{\mathbf{t}_1}$ is nonzero by construction in the initialization of the restarted portion of the proposed DEIM extension.

Now, suppose that $\hat{\mathbf{P}}_{j-1}^T \hat{\mathbf{V}}_{\mathbf{t}_{j-1}}$ is nonsingular, and let

$$\mathbf{r}_1 = \hat{\mathbf{v}}_\ell - \hat{\mathbf{V}}_{\mathbf{t}_{j-1}} (\hat{\mathbf{P}}_{j-1}^T \hat{\mathbf{V}}_{\mathbf{t}_{j-1}})^{-1} \hat{\mathbf{P}}_{j-1}^T \hat{\mathbf{v}}_\ell$$

for $\ell = \max(\mathbf{t}_{j-1}) + 1$. Notice that if $\|\mathbf{r}_1\|_\infty = 0$, then $\|\hat{\mathbf{r}}\|_\infty = 0$ for $\hat{\mathbf{r}}$ given by Eq. (2). In this case, $\hat{\mathbf{p}}_{j-1}$ and $\mathbf{t}_{j-1}$ are not updated.

Suppose, then, that $\|\hat{\mathbf{r}}\|_\infty > 0$. Then

$$\begin{aligned} 0 < \|\mathbf{r}_1\|_\infty &= |\mathbf{r}_1(\hat{p}_j)| = |\mathbf{e}_{\hat{p}_j}^T \mathbf{r}_1| \\ &= |\mathbf{e}_{\hat{p}_j}^T \hat{\mathbf{v}}_\ell - \mathbf{e}_{\hat{p}_j}^T \hat{\mathbf{V}}_{\mathbf{t}_{j-1}} (\hat{\mathbf{P}}_{j-1}^T \hat{\mathbf{V}}_{\mathbf{t}_{j-1}})^{-1} \hat{\mathbf{P}}_{j-1}^T \hat{\mathbf{v}}_\ell|, \end{aligned} \tag{3}$$

and $\mathbf{t}_j$ and $\hat{\mathbf{p}}_j$ are updated so that $\mathbf{t}_j = [\mathbf{t}_{j-1}; \ell]$ and $\hat{\mathbf{p}}_j = [\hat{\mathbf{p}}_{j-1}; \hat{p}_j]$. Factoring, we see that

$$\begin{aligned} \hat{\mathbf{P}}_j^T \hat{\mathbf{V}}_{\mathbf{t}_j} &= \begin{bmatrix} \hat{\mathbf{P}}_{j-1}^T \hat{\mathbf{V}}_{\mathbf{t}_{j-1}} & \hat{\mathbf{P}}_{j-1}^T \hat{\mathbf{v}}_\ell \\ \mathbf{e}_{\hat{p}_j}^T \hat{\mathbf{V}}_{\mathbf{t}_{j-1}} & \mathbf{e}_{\hat{p}_j}^T \hat{\mathbf{v}}_\ell \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I}_{j-1} & \mathbf{0} \\ \mathbf{e}_{\hat{p}_j}^T \hat{\mathbf{V}}_{\mathbf{t}_{j-1}} (\hat{\mathbf{P}}_{j-1}^T \hat{\mathbf{V}}_{\mathbf{t}_{j-1}})^{-1} & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{P}}_{j-1}^T \hat{\mathbf{V}}_{\mathbf{t}_{j-1}} & \hat{\mathbf{P}}_{j-1}^T \hat{\mathbf{v}}_\ell \\ \mathbf{0} & \nu_j \end{bmatrix} \end{aligned}$$

where $\nu_j = \mathbf{e}_{\hat{p}_j}^T \hat{\mathbf{v}}_\ell - \mathbf{e}_{\hat{p}_j}^T \hat{\mathbf{V}}_{\mathbf{t}_{j-1}} (\hat{\mathbf{P}}_{j-1}^T \hat{\mathbf{V}}_{\mathbf{t}_{j-1}})^{-1} \hat{\mathbf{P}}_{j-1}^T \hat{\mathbf{v}}_\ell \neq 0$ by Eq. (3).

Notice, then, that $\hat{\mathbf{P}}_j^T \hat{\mathbf{V}}_{\mathbf{t}_j}$ is the product of two nonsingular matrices. Hence, $\hat{\mathbf{P}}_j^T \hat{\mathbf{V}}_{\mathbf{t}_j}$ is itself nonsingular. □

For the purposes of demonstration, we are only considering the case where $\mathbf{p}$ has a length of up to $2k$ in this work, where the extent to which the length of $\mathbf{p}$ can extend beyond k elements is dependent upon the rank of $\hat{\mathbf{V}}$ in restarting DEIM one time. However, it is possible to restart DEIM multiple times in effort to select even more indices, still taking into account the rank of each subsequent matrix after the previously selected rows have been removed. To account for the selection of even more indices, we present a generalized algorithm of our extended DEIM approach in Algorithm 2 for a non-specific formulation of $\mathbf{r}_2$ with an added user-defined parameter, β, corresponding to the maximum number of indices to select. Note that the intended purpose in the presentation of Algorithm 2 is to convey the ideas behind the described extended DEIM approach; as written, Algorithm 2 is not very practical for implementation.

Algorithm 2 Extended DEIM (E-DEIM)

Input: $\mathbf{V}$, a full rank matrix in $\mathbb{R}^{m\times k}$ with $m > k$
τ, a positive real number for determining if an index should be added to $\mathbf{p}$
β, an integer such that $k \le \beta \le m$ indicating the maximum number of indices to include in $\mathbf{p}$

Output: $\mathbf{p}$, a vector in $\mathbb{N}^{\hat{k}}$ containing integral values from $\{1, ..., m\}$ with $k \le \hat{k} \le \beta$

% Perform standard DEIM
$\mathbf{v} = \mathbf{v}_1$
$[\sim, p_1] = \max(|\mathbf{v}|)$
$\mathbf{p} = p_1$
for $j = 2 : k$ **do**
  $\mathbf{v} = \mathbf{v}_j$
  $\mathbf{c} = \mathbf{V}(\mathbf{p}, 1 : j-1)^{-1}\mathbf{v}(\mathbf{p})$
  $\mathbf{r} = \mathbf{v} - \mathbf{V}(:, 1 : j-1)\mathbf{c}$
  $[\sim, p_j] = \max(|\mathbf{r}|)$
  $\mathbf{p} = [\mathbf{p}; p_j]$
end for

% Form the residual vector, $\mathbf{r}_2$, based on $\mathbf{V}(\mathbf{p}, :)$ and restart DEIM while $\mathbf{p}$ has $< \beta$ entries
while length$(\mathbf{p}) < \beta$ **do**
  $\mathbf{b} = \mathbf{p}^c$
  $\hat{\mathbf{V}} = \mathbf{V}(\mathbf{b}, :)$
  Compute $\mathbf{r}_2$ using chosen method

% Initialize the DEIM restart
  $\rho = 0$
  $i = 1$
  while $\rho \le \tau$ **and** $i \le m$ **do**
    $\hat{\mathbf{v}} = \hat{\mathbf{v}}_i$
    $[\rho, \hat{p}_1] = \max(|\hat{\mathbf{v}}. * \mathbf{r}_2|)$
    $i = i + 1$
  end while

  if $\rho \le \tau$ **and** $i == m + 1$ **then**
    break *% No rows contribute enough information to add more indices*
  else
    $\hat{\mathbf{p}} = \hat{p}_1$
    $\mathbf{t} = i - 1$

  % Search for additional DEIM indices
    while $i \le k$ **and** length$(\mathbf{t}) < (\beta -$ length$(\mathbf{p}))$ **do**
      $\hat{\mathbf{v}} = \hat{\mathbf{v}}_i$
      $\hat{\mathbf{c}} = \hat{\mathbf{V}}(\hat{\mathbf{p}}, \mathbf{t})^{-1}\hat{\mathbf{v}}(\hat{\mathbf{p}})$
      $\mathbf{r}_1 = \hat{\mathbf{v}} - \hat{\mathbf{V}}_{\mathbf{t}}\hat{\mathbf{c}}$
      $\hat{\mathbf{r}} = \mathbf{r}_1. * \mathbf{r}_2$
      $[\rho, \hat{p}_i] = \max(|\hat{\mathbf{r}}|)$

  % Update $\hat{\mathbf{p}}$ and $\mathbf{t}$ only if $\hat{\mathbf{v}}_i$ and row $\hat{p}_i$ both contribute enough new information
      if $\rho > \tau$ **then**
        $\hat{\mathbf{p}} = [\hat{\mathbf{p}}; \hat{p}_i]$
        $\mathbf{t} = [\mathbf{t}; i]$
      end if
      $i = i + 1$
    end while
  end if

  $\mathbf{p} = [\mathbf{p}; \mathbf{b}(\hat{\mathbf{p}})]$

end while

3.2 Further discussion of method

Before evaluating the performance of E-DEIM on data, we more thoroughly discuss the presented extension approach, focusing on the scenario with $\hat{k} \le 2k$ requiring only one restart of DEIM (although the discussed ideas can be adapted to apply to additional

restarts, as well). While there are a number of ways to select the additional indices included in extending $\mathbf{p}$, the approach shown in Algorithm 2 selects additional rows that span all of $\mathbb{R}^{\hat{k}-k}$ (as opposed to a subspace of $\mathbb{R}^{\hat{k}-k}$). Despite the fact that we lose the orthonormality, and perhaps even linear independence, of the columns in the original $\mathbf{V}$ when forming $\hat{\mathbf{V}} \in \mathbb{R}^{(m-k)\times k}$, ensuring that the $\mathbf{r}_1$ residual computed in line 30 of Algorithm 2 is large enough guarantees that the columns of $\hat{\mathbf{V}}(:, \mathbf{t})$ are at least linearly independent.

While having a more broadly spanning subset of rows is desirable, we would also like to make sure that the newly selected subset is different enough from the first subset selected with standard DEIM. The incorporation of the "memory" residual held in $\mathbf{r}_2$ adds extra computational cost, but serves as a bridge between the standard and restarted DEIM steps as well as a bridge between any subsequent DEIM restarts.

For the ℓ_1 formulation of $\mathbf{r}_2$, the computation of each entry in the $\mathbf{r}_2$ vector is $O(k^2)$: computing the ℓ_1-norm of the difference between the p_ℓth row of $\mathbf{V}$ and row η of $\hat{\mathbf{V}}$ is an $O(k)$ operation, then performing this for all k rows of $\mathbf{V}(\mathbf{p}, :)$ results in the $O(k^2)$ computational cost. Since the minimization is linear in k, then we can focus on higher order terms to approximate that performing this operation $m - k$ times (for each row in $\hat{\mathbf{V}}$) makes the computation of $\mathbf{r}_2$ with the ℓ_1 norm an $O(mk^2 - k^3)$ operation.

For the coherence-form of $\mathbf{r}_2$, once the rows of $\hat{\mathbf{V}}$ and $\mathbf{V}(\mathbf{p}, :)$ are normalized (with complexity $O(mk)$ and $O(k^2)$, respectively), the matrix–matrix multiplication to form $\mathbf{r}_2$ is $O(mk^2)$. Then, including normalization, computing $\mathbf{r}_2$ with the notion of coherence is $O(mk^2 + mk + k^2)$. (Note that the allowance of multiple restarts will increase these $\mathbf{r}_2$ computational costs according to the achieved increase in the length of $\mathbf{p}$ beyond k entries prior to each restart.)

Despite the added cost of both approaches to $\mathbf{r}_2$ presented above, the inclusion of this residual serves an important role in reducing the redundancy in class selection and, in the case where $k < \hat{k} \leq n$, is cheaper than re-computing a rank-$\hat{k}$ SVD of $\mathbf{A}$—an $O(mn\hat{k})$ operation. In addition, the ability to use different distance measures in computing $\mathbf{r}_2$ allows for flexibility in analyzing different data types; where we have discussed the use of the ℓ_1 distance on the rows of $\mathbf{V}$, other measures of distance may be more appropriate for some data sets (although calculating these distances may increase computational complexity). Here, we have chosen to compute $\mathbf{r}_2$ using rows of $\mathbf{V}$ as these rows may be shorter than those of $\mathbf{A}$ while still allowing us to leverage some of the underlying structure in the data. However, it is possible that there are settings in which it might be beneficial to instead compute $\mathbf{r}_2$ using the rows of the original data matrix $\mathbf{A}$ with other distance measures (such as dynamic time warping); this is a topic of future interest.

In theory, requiring the product of $\mathbf{r}_1$ and the normalized $\mathbf{r_2}$ to be above a certain tolerance forces the newly selected rows to be simultaneously linearly independent among themselves and different from the previously selected set. Hence, for the presented algorithm, we would expect to find that the additional rows selected via extended DEIM contain a subset of data points even more broadly representative than those selected via standard DEIM alone.

With this understanding of our approach, we briefly turn now to the application of an extended DEIM algorithm to the CUR factorization. Additional theoretical results in extending DIEM follow naturally from related works and are presented in "Appendix 1"; these results include a general bound on the extended DEIM projection error $\|\mathbf{A} - \mathcal{P}\mathbf{A}\|_2$. Again, while such bounds are clearly valuable in many contexts, the practical implications of such a bound in the class-identification setting is unclear and is an area of further interest. Nevertheless, we include these theoretical results for completeness.

3.3 Application to the CUR matrix factorization

As mentioned above, one of the uses of DEIM demonstrated in the existing literature is in constructing a CUR matrix factorization of the matrix $\mathbf{A}$, allowing for the identification of both representative rows and representative columns of a given data matrix while also forming an approximation to the original matrix with reduced dimensionality. In contrast to PCA, for instance, where the interpretability of the matrix factors is lost with respect to the original context, the matrices resulting from the CUR factorization contain rows/columns that maintain the original data structure and interpretation. Hence, the CUR factorization may be of interest in the scenario that both representative observations and representative features are desired from a large data set.

More specifically, a CUR matrix factorization is such that $\mathbf{A} \approx \mathbf{CUR}$ with $\mathbf{C} = \mathbf{A}(:, \mathbf{q})$ and $\mathbf{R} = \mathbf{A}(\mathbf{p}, :)$ for index vectors $\mathbf{p}, \mathbf{q} \in \mathbb{N}^k$, and $\mathbf{U}$ is defined such that the approximation holds—for example, we can set $\mathbf{U} = \mathbf{C}^\dagger \mathbf{A} \mathbf{R}^\dagger$ for $\mathbf{C}^\dagger$ and $\mathbf{R}^\dagger$ equal to the left and right pseudoinverses of $\mathbf{C}$ and $\mathbf{R}$, respectively. More details regarding the formation of such factorizations can be found in a number of papers such as those by Stewart (1999), Goreinov et al. (1997), Drineas et al. (2008), and Sorensen and Embree (2016), with our approach aligning closely with the DEIM-induced factorization described by Sorensen and Embree (2016). Where standard DEIM has been used to form $\mathbf{q}$ and $\mathbf{p}$ in the past, here we use E-DEIM to identify $\mathbf{q} \in \mathbb{N}^{\hat{k}}$ and $\mathbf{p} \in \mathbb{N}^{\hat{k}}$. Once again, in this context, using E-DEIM allows us to select more indices than allowed by the rank of the SVD, identifying larger row and/or column subsets that might be more informative given the application at hand. In the case where computing the SVD is prohibitively expensive, our method allows additional indices to be selected using only a low-rank SVD approximation without computing the SVD approximation for larger k.

Let

$$\mathrm{Q} = \mathrm{I}(:, [\tilde{\mathrm{q}}; \quad \hat{\mathrm{q}}]) = \begin{bmatrix} \tilde{\mathrm{Q}} & \hat{\mathrm{Q}} \end{bmatrix}$$

and

$$\mathrm{P} = \mathrm{I}(:, [\,\tilde{\mathrm{p}}; \quad \hat{\mathrm{p}}]) = \begin{bmatrix} \tilde{\mathrm{P}} & \hat{\mathrm{P}} \end{bmatrix}$$

where $\tilde{\mathbf{q}}$ and $\tilde{\mathbf{p}}$ are those indices selected via standard DEIM and $\hat{\mathbf{q}}$ and $\hat{\mathbf{p}}$ are the additional indices selected through the presented extension of DEIM.

Using the subscript notation presented in Lemma 1, we use the fact that $\hat{\mathbf{P}}_j^T \hat{\mathbf{V}}_{\mathbf{t}_j}$ is invertible for each $1 \leq j \leq (\hat{k} - k)$ to conclude that $\hat{\mathbf{P}}^T \hat{\mathbf{V}}$ has full row rank. Then we can also conclude that our selection of an additional $\hat{k} - k$ linearly independent rows from $\mathbf{V}$ yields a submatrix $\mathbf{A}(\mathbf{p}(k+1 : \hat{k}), :)$ that has full row rank. In a similar manner, we can select an additional set of linearly independent columns $\mathbf{A}(:, \mathbf{q}(k+1 : \hat{k}))$.

For simplicity—and for its relevance to the observation subset selection from a matrix containing observations arranged in a column-wise manner—for now, we consider only the case with $\mathbf{q} \in \mathbb{N}^{\hat{k}}$ and keep $\mathbf{p} \in \mathbb{N}^k$. Using extended DEIM to select columns and standard DEIM to select rows of the rank-r matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ via its rank-k SVD, we can still obtain a CUR decomposition of the form $\mathbf{A} \approx \mathbf{CUR}$ where we now have $\mathbf{C} \in \mathbb{R}^{m \times \hat{k}}$, $\mathbf{U} \in \mathbb{R}^{\hat{k} \times k}$, and $\mathbf{R} \in \mathbb{R}^{k \times n}$. Suppose $\mathbf{C}$ has rank κ for $k \leq \kappa \leq \min\{\hat{k}, r\}$. Without loss of generality, suppose that the first κ columns of $\mathbf{C}$ are linearly independent with the first k columns selected via standard DEIM and the next $(\kappa - k)$ columns selected via an extension, and suppose the remaining $(\hat{k} - \kappa)$ columns lie in the span of the first κ columns. Then we can write

$$\mathbf{A} \approx \mathbf{CUR} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2 \end{bmatrix} \begin{bmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{bmatrix} \mathbf{R}, \tag{4}$$

where $\mathbf{C}_1 \in \mathbb{R}^{m\times\kappa}$ and $\mathbf{C}_2 \in \mathbb{R}^{m\times(\hat{k}-\kappa)}$ both have full column rank but $\text{Range}(\mathbf{C}_2) \subset \text{Range}(\mathbf{C}_1)$, and $\mathbf{U}_1 \in \mathbb{R}^{\kappa\times k}$ and $\mathbf{U}_2 \in \mathbb{R}^{(\hat{k}-\kappa)\times k}$.

If the E-DEIM-selected columns of $\mathbf{C}_2$ are unknown, we can set

$$\mathbf{U_1} = \begin{bmatrix} \mathbf{C}_1(:, 1:k)^{\dagger}\mathbf{A}\mathbf{R}^{\dagger} \\ \mathbf{0} \end{bmatrix}$$

and $\mathbf{U}_2 = \mathbf{0}$. This is the same structure of $\mathbf{U}$ that can be used for $\kappa = k < \min\{\hat{k}, r\}$; we can take $\mathbf{U}_1 = \mathbf{C}_1^{\dagger}\mathbf{A}\mathbf{R}^{\dagger}$ and define $\mathbf{U}_2 = \mathbf{0}$. With this choice of $\mathbf{U}$, we can then find an approximation error bound for

$$\|\mathbf{A} - \mathbf{CUR}\| = \|\mathbf{A} - \mathbf{C}_1\mathbf{U}_1\mathbf{R}\|,$$

where $\|\cdot\|$ is the induced matrix 2-norm. Since $\mathbf{C}_1$ and $\mathbf{R}$ simply contain the standard-DEIM-selected indices, the CUR error bound follows directly from Theorem 4.1 by Sorensen and Embree (2016). With $\mathbf{C}_1 = \mathbf{AQ}(:, 1:k)$, this proof only requires use of the first k columns of $\mathbf{Q}$, denoted as $\tilde{\mathbf{Q}}$ above, with $\mathbf{W}^T\tilde{\mathbf{Q}}$ invertible.

Suppose, however, that $\kappa > k$ and the E-DEIM-selected columns are known so that we may define $\mathbf{U}_1 = \mathbf{C}_1^{\dagger}\mathbf{A}\mathbf{R}^{\dagger}$, which is not a square matrix, and $\mathbf{U}_2 = \mathbf{0}$. With more detail provided in "Appendix 1", and with $\mathrm{Q}_1 = \mathrm{I}(:,\mathrm{q}_1)$ such that $\mathrm{AQ}_1 = \mathrm{C}_1$, we state the following approximation error bound for CUR with an extension of DEIM, paralleling Theorem 4.1 presented by Sorensen and Embree (2016).

Theorem 1 *Suppose* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *and* $k \le rank(A)$ *with* $1 \le k < \kappa \le \min\{m, n\}$. *Let* $\mathbf{C}_1 = \mathbf{A}(:, \mathbf{q}_1) \in \mathbb{R}^{m\times\kappa}$, $\mathbf{R} = \mathbf{A}(\mathbf{p}, :) \in \mathbb{R}^{k\times n}$ *and* $\mathbf{U}_1 = \mathbf{C}_1^{\dagger}\mathbf{A}\mathbf{R}^{\dagger}$. *Then*

$$\|\mathrm{A} - \mathrm{C}_1\mathrm{U}_1\mathrm{R}\| \le (\|(\mathrm{P}^T\mathrm{V})^{-1}\| + \|(\mathrm{W}^T\mathrm{Q}_1)^{\dagger}\|)\sigma_{k+1}. \tag{5}$$

In the case where the CUR factorization is defined as in Eq. (4) with $\mathbf{U}_2 = \mathbf{0}$, the bound in (5) is the same for $\|\mathbf{A} - \mathbf{CUR}\|$. We also note that where this result is presented for the case where only $\mathbf{C}$ contains E-DEIM-selected columns that might contribute additional information regarding the column space of $\mathbf{A}$, very similar results can be obtained for the cases in which E-DEIM is used to form both $\mathbf{C}$ and $\mathbf{R}$ or only $\mathbf{R}$.

These results presented in this section suggest that extensions of DEIM that minimize $\|(\mathbf{W}^T\mathbf{Q}_1)^{\dagger}\|$ and $\|(\mathbf{P}^T\mathbf{V})^{-1}\|$ are more theoretically desirable. While we do not currently have a bound on such quantities for our methods, this is a topic of future interest. Zimmermann and Willcox (2016) and Peherstorfer et al. (2018) present oversampling techniques designed to reduce these errors. Although this may be preferred in some settings, the methods in these other works have the potential to be more computationally expensive than those developed here (depending on the matrix properties and the choice in computing $\mathbf{r}_2$). In addition, while discussed here for a broader picture regarding the incorporation of DEIM extensions in CUR, improved error bounds may not translate as well to some machine learning tasks since successful class identification in relation to these bounds is not yet well understood.

4 Description of data experiments

We next evaluate the performance of extended DEIM (E-DEIM) with one restart in the context of subset selection. For comparison with the use of standard DEIM in the ECG analysis setting (Hendryx et al., 2018), we apply E-DEIM to the MIT-BIH Arrhythmia Database (Moody & Mark, 2001) available in PhysioNet (Goldberger et al., 2000). To highlight the role of E-DEIM in identifying more classes than allowed by the rank of the corresponding data matrix, we also apply our approach to the Letter Recognition Data Set (Frey & Slate, 1991) from the UCI Machine Learning Repository (Dua & Taniskidou, 2017). For each data set, we compare the performance of DEIM and E-DEIM with results from applying k-medoids clustering and statistical leverage scores in identifying representative subsets. Our implementations of these more commonly used techniques are described further below.

4.1 MIT-BIH Arrhythmia Database

The MIT-BIH Arrhythmia Database consists of 48 files containing 30-min two-lead electrocardiogram (ECG) recordings from 47 different patients (Moody & Mark, 2001). Each waveform is presented in mV in PhysioNet (Goldberger et al., 2000)) with three digits of accuracy. Although ways for including additional leads in subset selection are suggested by Hendryx et al. (2018), we only consider one lead in our analyses; the MLII lead is processed when available, with the V5 lead analyzed in cases without access to the MLII lead. These MIT-BIH recordings contain whole-beat annotations that serve as a reference here for determining whether or not all of the different expected beat classes are detected. As is done in a number of works in the literature (for example, De Chazal et al., 2004), the files containing paced beats are removed from consideration and the remaining data set is divided into a training set and a test set—called DS1 and DS2, respectively—each with 22 files. (The specific DS1/DS2 data split used by De Chazal et al. (2004) is included in "Appendix 2" for reference.) In constructing the data matrix for each file, the waveform is first preprocessed via a zero-phase first order high pass Butterworth filter with a $5 \times 10^{-3}\pi$ radians-per-second cutoff frequency in effort to reduce baseline wandering. A conservative 5% of the full signal length is removed from each end of the waveform to eliminate edge effects from filtering. This filtered and trimmed data is then divided into individual RR-intervals using the RR-interval data provided with the waveforms in PhysioNet (Goldberger et al., 2000)). Each RR-interval is interpolated to contain 150 time samples and then normalized to have a mean of zero and standard deviation of one. These beats defined from R-peak to R-peak are used to construct a data matrix with each column corresponding to the amplitudes of an individual beat.

Each beat (or column) in the file data matrix is also assigned a label corresponding to the physician-given annotation included with the data set at the RR-interval onset. Given the provided labels, we consider the following annotation classes within the data: normal beat (N), left bundle branch block beat (L), right bundle branch block beat (R), atrial premature beat (A), aberrated atrial premature beat (a), nodal (junctional) premature beat (J), supraventricular premature or ectopic beat (S), premature ventricular contraction (V), fusion of ventricular and normal (F), atrial escape beat (e), nodal (junctional) escape beat (j), ventricular escape beat (E), and unclassifiable beat (Q) (Goldberger et al., 2000). Of note is that the beats with labels E, e, and S only appear in the training set as they each

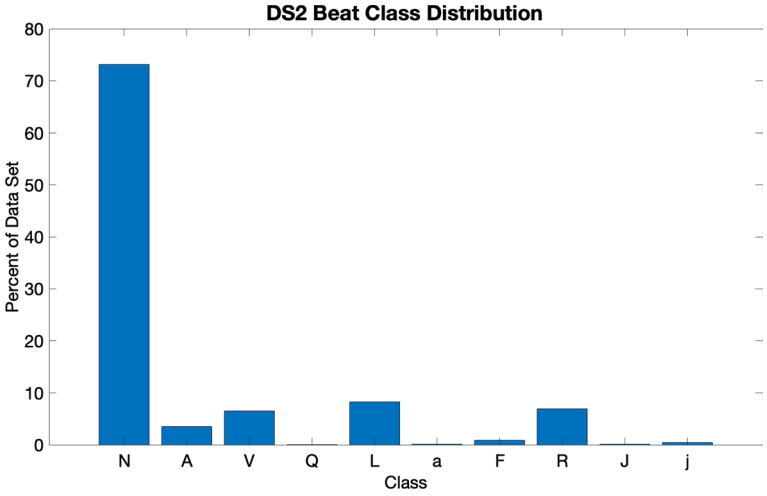

Fig. 1 The distribution of class labels among the beats in the DS2 subset of the MIT-BIH Arrhythmia Database tested here

only appear in one file allocated to the DS1 data subset; hence, the training and test sets do not have the same expected number of classes to detect. The distribution of the classes in the DS2 test set is shown in Fig. 1. Form this figure, we can see that the classes are not uniformly distributed, and some annotations only appear a few times in the data set. As is done by Hendryx et al. (2018), we also include results for evaluating our methods in ignoring class detection (or lack thereof) within a file when fewer than three beats are present with the corresponding label; in this way, we can also see how each of the evaluated methods perform when the detection of those extremely rare-occurring beats is of low priority.

Examples of individual beats stored in the training data matrix for patient file numbers 101 and 124 are shown in Fig. 2 along with their corresponding beat labels. Notice that while some annotations are repeated, the morphologies corresponding to such annotations may vary within the given class; for example, notice that the third beat in the top row in Fig. 2 carries an annotation of "N" even though it is notably different from the other RR-intervals labeled with "N" (normal beat). Such within-class variability holds potential for multiple representatives to be selected from a given class in an unsupervised learning task, as we will observe in Sect. 5. Depending on the application at hand, this higher-resolution class identification in the machine learning context may be a desirable outcome.

4.2 Letter Recognition Data Set

The Letter Recognition Data Set consists of 20, 000 observations of 16 features derived from perturbed images of letters from the English alphabet in a variety of fonts. Each image feature has been scaled to take on integer values from 0 to 15 (Frey & Slate, 1991); since the data is already scaled to have values within a fixed range, we do not normalize this data prior to analysis. Each of the 26 classes is well-represented in the data set; by "well-represented," we mean that each of the 26 classes has > 700 observations. Splitting the data randomly into training and test sets of equal sizes, each of the corresponding $16 \times 10{,}000$ data matrices are prime candidates for class detection through column selection with an extension of DEIM. Where we would hope to identify representatives for each of the 26 letters

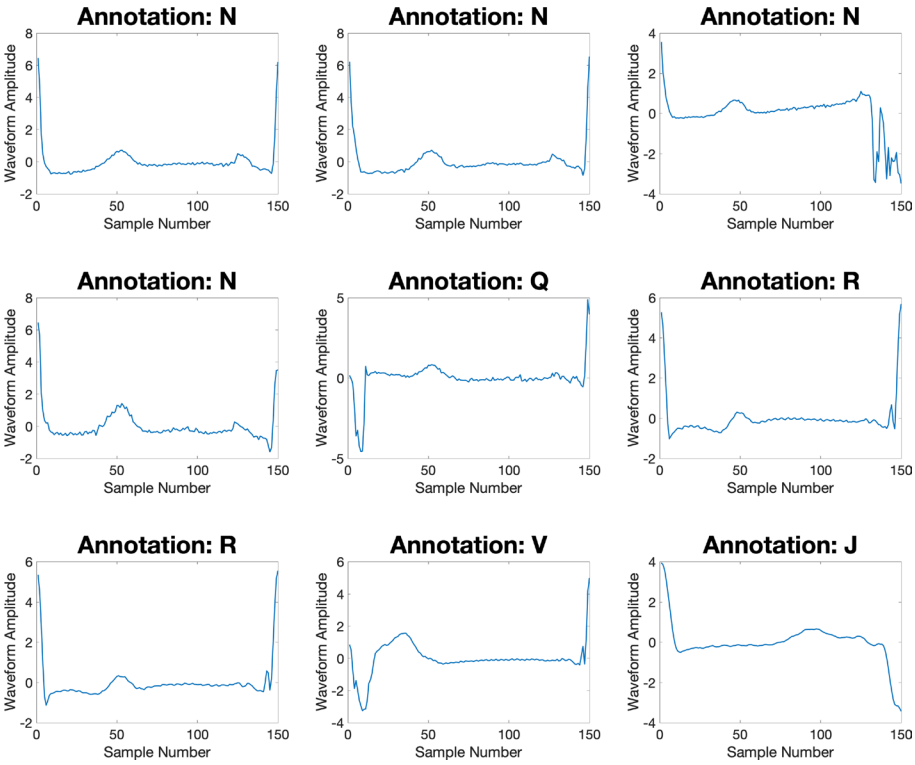

Fig. 2 Examples of beat observations in the data matrices constructed from the MIT-BIH Arrhythmia Database files corresponding to patient file numbers 101 and 124

in the English alphabet, the fact that each of the corresponding data matrices has a rank of at most 16 makes it impossible for standard DEIM to detect all 26 classes in both training and test sets. Therefore, the only way to identify all 26 classes with a DEIM-like method is to make use of extension/oversampling methods such as those presented here.

4.3 Evaluated methods

Subset selection is performed on both of the data sets described above using seven different approaches: five approaches stem from the DEIM index-selection scheme, one approach makes use of the *k*-medoid clustering algorithm, and the seventh approach uses leverage scores. Each approach is described further below.[2]

[2] Initial work comparing standard DEIM with additional methods and including application to different data sets can be found in the dissertation by Hendryx (2018) as work completed with Nabil Chaabane.

4.3.1 Variations on E-DEIM

To better understand the role of the different pieces of the proposed E-DEIM algorithm discussed above, we use DEIM to select a representative subset in each of these data sets with five different DEIM-type implementations: standard DEIM by itself, standard DEIM with additional indices randomly selected, E-DEIM considering only $\mathbf{r}_1$ and no "memory" residual (i.e. $\hat{\mathbf{r}} = \mathbf{r}_1$), E-DEIM with the ℓ_1 distance in computing $\mathbf{r}_2$, and E-DEIM with $\mathbf{r}_2$ capturing a sense of coherence. In the results below, these will be denoted as 'DEIM$_S$', 'DEIM$_{rand}$', 'E-DEIM$_{\mathbf{r}_1}$', 'E-DEIM$_{\ell_1}$', and 'E-DEIM$_{coh}$', respectively.

In generating results for DEIM$_{rand}$, the additional indices are randomly selected without replacement from a uniform distribution using MATLAB's default random number generator, and the number of indices selected is chosen to match the number of indices selected by the other extension methods. The DEIM$_{rand}$ experiment is repeated 100 times, and it is the average of these results that is reported.

DEIM/E-DEIM parameter selection

With standard DEIM and the three E-DEIM approaches applied to each training set, parameter selection is performed for the SVD truncation tolerance, θ, to determine the rank, k, of the SVD approximation. The value of k corresponding to each θ-value is such that k is the smallest index with $\sigma_k/\sigma_1 > \theta$ for singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k \geq \sigma_n$. Parameter selection is performed from the θ-values of $\theta = 0.5,\ 0.1,\ 5 \times 10^{-2},\ 10^{-2},\ 5 \times 10^{-3}, 10^{-3}, 5 \times 10^{-4}, 10^{-4}, 5 \times 10^{-5}$, and 10^{-5}.

For the extended DEIM approaches, the extension tolerance within the restarted portion of DEIM, denoted as τ above, is also selected on the respective training sets from among the parameter values of $\tau = 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}, 10^{-9}$, and 10^{-10}. While it is not really necessary to tune this parameter if the goal is solely to avoid a nearly singular matrix, larger τ-values can also potentially play a role in the number of indices added to $\mathbf{p}$ in each restart. The tested values here are chosen more for the sake of avoiding a nearly singular matrix without accounting for the digits of accuracy in the original data sets; in future applications, parameter selection should also consider the accuracy of the data sets of interest and explore the potential role of τ in defining a stopping criteria for the number of DEIM restarts to perform.

As there are at most two hyperparameters to select in applying E-DEIM with one restart (fixing $\beta = 2k$), and we are not otherwise fitting additional model parameters for class identification, we perform model selection by assessing the performance of each (θ, τ) parameter pair on the training set without cross-validation or a separate validation set. For both θ and τ, the final parameter selection for the MIT-BIH data set is given by the largest parameter values obtaining the maximum amount of annotation detection across all files and all annotations in the training set, DS1; that is, the parameter pair is chosen to achieve the highest detection rates in answering the following question: Of all individual labels to be detected on a file-by-file basis, how many of those labels are selected via the given DEIM/E-DEIM implementation? Similarly, for the Letter Recognition Data Set, the final parameter combination of θ and τ is chosen to minimize the number of missed classes in the training set. For both data sets, the chosen parameter pair is selected to simultaneously maximize performance for DEIM and all of the tested E-DEIM extensions, resulting in the same rank-k SVD approximation for all experiments that require the use of the singular vectors. After applying DEIM and E-DEIM to the test data sets with the selected parameters, we can then compare the test set results

with those produced from applying DEIM$_{rand}$ along with k-medoids clustering and leverage score row/column selection (described below) under the same amount of dimension reduction.

4.3.2 *k*-medoids clustering

Similar to the more prominent k-means algorithm, k-medoids clustering is a well-known unsupervised learning technique that partitions the data into k groups, each group centered around a particular point in the data set (or "medoid"). We can consider these medoids to be data class representatives, and therefore, we can use the medoids to select a subset of the data. It is worth noting, however, that there is usually some form of randomness in selecting the algorithm's starting points. Hence, this clustering approach does not necessarily produce the same results each time it is applied to a data set.

For our experiments, we use the implementation of k-medoids available in MATLAB_R2020a. In particular, the underlying selected algorithm is based on that proposed by Park and Jun (2009) and is implemented without calling for the optional online update phase that carries out an extra iteration similar to the classic PAM (Partitioning Around Medoids) algorithm. While the inclusion of a PAM-like iteration holds the potential to improve the clustering results (MATLAB, 2020), PAM has an $O(k(m-k)^2)$ computational cost compared to the cheaper $O(mk)$ cost of running Park's and Jun's algorithm by itself (Park and Jun, 2009); while the MATLAB documentation states that the "PAM-like" update iteration is typically computationally cheaper than an actual PAM iteration (MATLAB, 2020), we choose to carry out clustering without this update phase. We average the results across ten different `kmedoids` function calls, where each individual function call also outputs the clustering result from selecting ten different sets of initial medoids via MATLAB's `k-means++` algorithm. The selected distance measure is the default squared Euclidean distance, and we fix the number of selected medoids, k, to match the number of indices selected via standard DEIM and E-DEIM for comparable dimension reduction. We denote the k-medoid results with k matching the number of DEIM-selected indices as 'k-Med$_D$', and we use 'k-Med$_E$' to denote the results for k matching the number of E-DEIM-selected indices.

4.3.3 Leverage scores

In addition to k-medoids clustering, we also compare DEIM and the extensions proposed herein to the commonly known column selection technique using statistical leverage scores. Also used to select columns and rows in constructing a CUR decomposition of $\mathbf{A}$ [see the works by Drineas et al. (2008) and Mahoney and Drineas (2009)], the leverage score approach uses a probability distribution from which to select indices. This distribution for row selection, for example, is formed by computing the normalized leverage scores based on the squared ℓ_2 -norms of each row in the matrix containing the singular vectors of $\mathbf{A}$. More specifically, the leverage score corresponding to the ith row of $\mathbf{A}$ is computed as

$$\pi_i = \frac{1}{k}\sum_{j=1}^{k} \mathbf{V}(i,j)^2,$$

where $\mathbf{V}$ is the matrix of the first k left singular vectors from the SVD of $\mathbf{A}$. We then define the index vector $\mathbf{p}$ to contain the indices of the rows with the k largest leverage scores. Some implementations automatically oversample the leverage scores to find more than

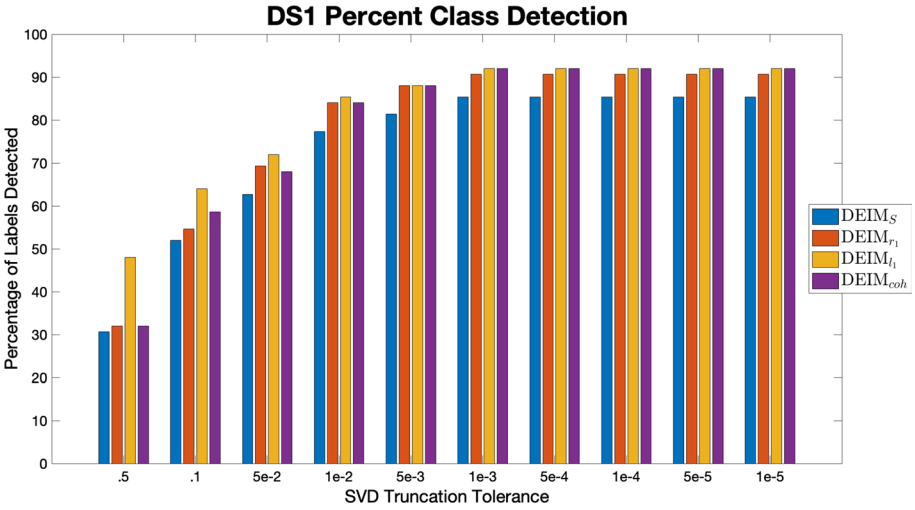

Fig. 3 Percent detection of all labels present among all files in the DS1 MIT-BIH Arrhythmia Database training set for the DEIM variations DEIM$_S$, E-DEIM$_{\mathbf{r}_1}$, E-DEIM$_{\ell_1}$, and E-DEIM$_{coh}$ and each of the ten tested SVD truncation tolerances. These results show, of all annotations to be detected on a file-by-file basis, how many of those individual annotations are detected through a particular variation of DEIM

k indices; we simply use the same rank-k SVD to select the same number of indices as selected by DEIM and E-DEIM for comparison. In the results reported below, we refer to the leverage score results with the number of selected indices matching that of DEIM as 'Lev$_D$' and the leverage score results with the number of selected indices matching that of E-DEIM as 'Lev$_E$'.

5 Results and discussion

With the data sets and parameter values discussed above, we now present the results from applying the different extensions of DEIM described herein along with the results of applying DEIM$_{rand}$, k-medoids clustering, and leverage score index selection.

5.1 MIT-BIH Arrhythmia Database results

To test the extended DEIM methods on the MIT-BIH data set, we conduct trials similar to those carried out in our previous work using standard DEIM (Hendryx et al., 2018), but we directly compute the rank-k SVD of the data matrix as opposed to finding an approximation using an incremental QR approach. To select the SVD truncation tolerance, θ, and the extension tolerance, τ, we consider the percent detection among all present labels across all files in DS1. In computing this percentage of label detection, the annotations in each DS1 file are considered independently; for example, if two out of three annotations are detected in one file and four out of five annotations are detected in another file, the detection summary for the two files would be six out of eight annotations, or 75%, even if some annotations are present in both files. These summary results on the training set are are shown in Fig. 3 for the different θ-values used. Interestingly, in all relevant cases, the detection

Table 1 Percent detection of all annotations to be detected on a file-by-file basis for the DS2 subset of the MIT-BIH Arrhythmia Database with $\theta = 10^{-3}$ and, in the extended methods, $\tau = 10^{-4}$

Standard	Full data	≥ 3 beats/label	Extended	Full data	≥ 3 beats/label
DEIM_S	90	94.55	DEIM_{rand}	91.84	95.96
$k\text{-Med}_D$	86.14	95.45	$\text{E-DEIM}_{\mathbf{r}_1}$	92.86	98.18
Lev_D	90	94.55	E-DEIM_{ℓ_1}	92.86	98.18
			E-DEIM_{coh}	94.29	100
			$k\text{-Med}_E$	92.57	98
			Lev_E	92.86	98.18

The left column holds results for $\mathbf{p}$ having length k, and the right column holds results for $\mathbf{p}$ having length $2k$

Table 2 The annotation detection results for the full DS2 subset of the MIT-BIH Arrhythmia Database with $\theta = 10^{-3}$ and $\tau = 10^{-4}$

DS2	N	A	V	Q	L	a	F	R	J	j
DEIM_S	100	84.62	100	100	100	100	57.14	100	50	50
$k\text{-Med}_D$	100	70.77	96.88	70	100	97.5	55.71	100	60	60
Lev_D	100	84.62	100	100	100	100	57.14	100	50	50
DEIM_{rand}	100	87.62	100	100	100	100	67.43	100	51.5	57.5
$\text{E-DEIM}_{\mathbf{r}_1}$	100	100	100	100	100	100	57.14	100	50	50
E-DEIM_{ℓ_1}	100	92.31	100	100	100	100	71.43	100	50	50
E-DEIM_{coh}	100	100	100	100	100	100	71.43	100	50	50
$k\text{-Med}_E$	100	81.54	98.13	100	100	97.5	85.71	100	70	60
Lev_E	100	92.31	100	100	100	100	71.43	100	50	50

results are insensitive to the choice of τ used in the restarted DEIM portion of the algorithm; hence the training detection results shown in Fig. 3 are only for the truncation tolerances of the SVD. The reason for this insensitivity to τ is likely that the remaining columns of $\hat{\mathbf{V}}$ are still linearly independent for the matrices associated with this data set, resulting in residual vectors that have entries greater than $\tau = 10^{-4}$.

From the results in Fig. 3, we see improved performance for standard DEIM and each of the proposed DEIM extensions as the SVD truncation tolerance decreases to $\theta = 10^{-3}$. At this value of θ, the resulting k-value is often close to the number of samples per beat (suggesting the corresponding file data matrices are nearly full rank), and the detection percentages for DEIM_S, $\text{E-DEIM}_{\mathbf{r}_1}$, E-DEIM_{ℓ_1}, and E-DEIM_{coh} are 85.33%, 90.67%, 92%, and 92%, respectively. Fixing $\tau = 10^{-4}$, we select a truncation tolerance of $\theta = 10^{-3}$ in applying each of the DEIM implementations to the test set, DS2. Given the resulting matrix rank from using the selected SVD truncation tolerance, the remaining methods (DEIM_{rand}, k-medoids, and leverage scores) are also applied to have the same number of indices selected as those in both DEIM (k) and E-DEIM ($2k$). The corresponding results are shown in Table 1. Here we present the detection summary results across all annotations and files for both the full data (with rare-occurring annotations) and the case in which the detection (or lack of detection) in annotations occurring few than three times is ignored in a given file. The left half of Table 1 contains the results

Table 3 The annotation detection results for annotations appearing ≥ 3 times in a file for the DS2 subset of the MIT-BIH Arrhythmia Database with $\theta = 10^{-3}$ and $\tau = 10^{-4}$

DS2	N	A	V	Q	L	a	F	R	J	j
DEIM$_S$	100	77.78	100	100	100	100	66.67	100	100	100
k-Med$_D$	100	78.89	100	90	100	97.5	86.67	100	100	100
Lev$_D$	100	77.78	100	100	100	100	66.67	100	100	100
DEIM$_{rand}$	100	82.11	100	100	100	100	79.67	100	100	100
E-DEIM$_{\mathbf{r}_1}$	100	100	100	100	100	100	66.67	100	100	100
E-DEIM$_{\ell_1}$	100	88.89	100	100	100	100	100	100	100	100
E-DEIM$_{coh}$	100	100	100	100	100	100	100	100	100	100
k-Med$_E$	100	88.89	100	100	100	97.5	100	100	100	100
Lev$_E$	100	88.89	100	100	100	100	100	100	100	100

for the tested methods having the same number of k selected indices, and the right half contains the results for all of the evaluated extensions with $2k$ selected indices.

An annotation-by-annotation breakdown of the results on the full DS2 data set is shown in Table 2, and the corresponding results focused solely on detection in files with annotations represented by three or more beats are shown in Table 3. In all three results tables, Tables 1, 2, and 3, we see that simply extending the method to restart DEIM improves the detection results. While this is perhaps not very surprising given that all of the presented DEIM extensions (including the addition of randomly selected indices) simply add to the standard DEIM results, it confirms that selecting additional indices does allow for even more classes to be identified—not just re-selection from those classes previously identified. Although the overall improvement over standard DEIM shown in Table 1 is the same for both E-DEIM$_{\mathbf{r}_1}$ and E-DEIM$_{\ell_1}$, Tables 2, and 3 demonstrate that the improvement is spread out across two classes (A and F) in E-DEIM$_{\ell_1}$ as opposed to the single class (F) with improved results for E-DEIM$_{\mathbf{r}_1}$. The greatest improvement seen in class detection over DEIM$_S$, however, is in the application of E-DEIM$_{coh}$ in which not only is DEIM restarted, but the "memory" residual, $\mathbf{r}_2$, is computed using a measure of coherence. Hence, we see that the incorporation of $\mathbf{r}_2$ in the extension of DEIM can add value to the overall algorithm performance in class detection.

In comparison to DEIM$_S$ and the proposed E-DEIM approaches, Table 1 shows that k-Med$_D$ performs worse than DEIM$_S$ when applied to the full data set but slightly better than DEIM$_S$ when the rare-occurring labels are ignored; k-Med$_E$ performs worse than all of the investigated extensions with the exception of DEIM$_{rand}$, performing only slightly worse than E-DEIM$_{\mathbf{r}_1}$ and E-DEIM$_{\ell_1}$. Tables 2 and 3 show the more detailed k-medoids results, showing that k-medoids clustering can lead to improved detection results over DEIM/E-DEIM in some specific classes, but worse detection results in others.

The leverage score approach, on the other hand, produces percentages that match the performance of DEIM$_S$ for the smaller index set and percentages that match the performance of E-DEIM$_{\ell_1}$ for the larger index set. The results are even matched on the class-by-class breakdown of the results in both Tables 2 and 3. The inclusion of randomly selected indices in DEIM$_{rand}$, however, appears the least beneficial in this setting. While we do see improvement over DEIM$_S$ by just selecting additional representatives, this improvement is lacking in comparison to the more structured extension approaches, especially E-DEIM$_{coh}$.

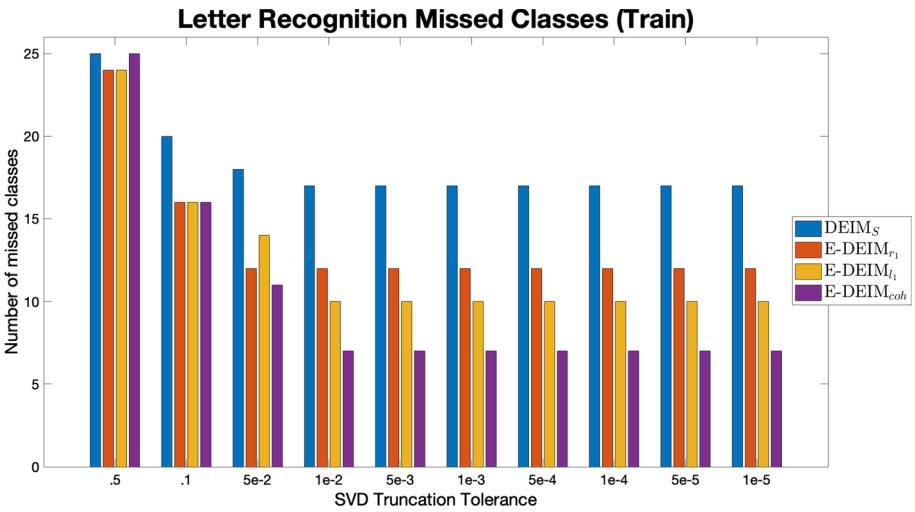

Fig. 4 Number of missed classes out of the 26 classes present in the Letter Recognition Data Set for each of the four DEIM variations, DEIM$_S$, E-DEIM$_{\mathbf{r}_1}$, E-DEIM$_{\ell_1}$, E-DEIM$_{coh}$ and each of the ten tested SVD truncation tolerances

With these improved results, it is of little surprise that the amount of dimension reduction decreases through extending DEIM. Where standard DEIM sees a dimension reduction of 92.99% from the original 44,664 beats, each of the extensions yields a reduction by 85.98% on the DS2 data set. This, however, is a direct result of constructing the algorithm to detect up to $2k$ class representatives rather than stopping at k. While some classes are over-represented in our selected subsets, as pointed out in looking at the sample beats in Fig. 2, the distinctions made between different morphologies within a given clinically-defined class may be of benefit in developing clinical decision support tools such as automated cardiac event prediction algorithms.

5.2 Letter Recognition Data Set results

Because the initial motivation in extending DEIM was to be able to handle class detection in the case where the number of classes is likely to exceed the rank of the matrix, we also test the performance of extended DEIM on the Letter Recognition Data Set described by Frey and Slate (1991) and available through the UCI Machine Learning Repository (Dua & Taniskidou, 2017). Again, since there are only 16 derived features for each letter image, the rank of the corresponding data matrix must be no greater than 16, making it necessary for the use of an extension if a DIEM-type approach is going to have a chance of identifying a set of representative features from all 26 classes (or letters). As with the MIT-BIH Arrhythmia Database, we perform parameter selection on the training half of this letter recognition data. The class detection results are shown in Fig. 4, displaying how many of the 26 letter classes are missed in the application of DEIM and each E-DEIM variation. Because these E-DEIM results are also insensitive to the choices of τ tested (the largest values in $\hat{\mathbf{r}}$ remain greater than the tested extension thresholds), we again only present the results for the corresponding SVD truncation tolerances. We see from these results that the number of

Table 4 Class detection and dimension reduction results for the test portion of the Letter Recognition Data Set

Standard	Missed classes	Dim. red. (%)	Extended	Missed classes	Dim. red. (%)
DEIM_S	13	99.84	DEIM_{rand}	6.91	99.68
$k\text{-Med}_D$	12.70	99.84	$\text{E-DEIM}_{\mathbf{r}_1}$	11	99.68
Lev_D	21	99.84	E-DEIM_{ℓ_1}	8	99.68
			E-DEIM_{coh}	7	99.68
			$k\text{-Med}_E$	6.80	99.68
			Lev_E	20	99.68

The left column holds results for $\mathbf{p}$ having length k, and the right column holds results for $\mathbf{p}$ having length $2k$

missed classes across all four evaluated methods generally decreases as θ decreases to 10^{-2}, where DEIM_S misses 17 classes, $\text{DEIM}_{\mathbf{r}_1}$ misses 12 classes, DEIM_{ℓ_1} misses ten classes, and DEIM_{coh} misses seven classes. Notice that for $\theta = 0.5$, DEIM_{coh} performs just as well as standard DEIM. It is worth noting here that for this truncation tolerance, the resulting low-rank SVD approximation has rank one; hence, any attempt to compute the coherence between the rows of $\mathbf{V} \in \mathbb{R}^{m\times 1}$ and $\hat{\mathbf{V}} \in \mathbb{R}^{(m-1)\times 1}$ will preclude the selection of an additional index in the restarted DEIM step due to the fact that elements of $\mathbb{R}^1$ are collinear.

Given the results from Fig. 4, we fix $\theta = 10^{-2}$ and $\tau = 10^{-4}$ in applying the different DEIM implementations to the letter recognition test set and obtain the results in Table 4. This table shows the number of missed classes for each tested method along with its corresponding dimension reduction. As with the MIT-BIH results in Table 1, the results on the left in Table 4 correspond to methods selecting only k indices and methods on the right correspond to the methods selecting $2k$ indices, where, for $\theta = 10^{-2}$, the resulting k-value is 16. From this table, we again see E-DEIM methods outperform standard DEIM in class detection. The role of the "memory" residual appears to be more important for class detection in this data set, with both E-DEIM_{ℓ_1} and E-DEIM_{coh} missing fewer classes than $\text{E-DEIM}_{\mathbf{r}_1}$.

In comparing DEIM and E-DEIM to the other methods discussed here, the results in Table 4 show that the use of leverage scores is ill-suited for this data set as the only classes identified for $k = 16$ correspond to the letters J, M, N, Y, and Z, and extending the method to allow for the selection of $2k = 32$ representatives yields only the added representation of the letter Q. k-medoids, on the other hand, performs slightly better (on average) than DEIM_S and E-DEIM_{coh}, with DEIM_{rand} producing average results that are between those of $k\text{-Med}_E$ and E-DEIM_{coh}. It is important to note, however, that all classes in this data set are well-represented; hence, the randomness in both the initialization of k-medoids and the selection of indices for DEIM_{rand} might actually help in class detection in this setting where it did not appear to help as much in analyzing the unevenly distributed ECG data. While we are able to compute the average number of missed classes for both of these techniques in generating the results for Table 4, a question that might follow from applying these repeated experiments in a truly unsupervised setting without knowledge of any labels is: How do we know which experiment's indices should be trusted? It is this question along with the $k\text{-Med}_E$ and DEIM_{rand} improved detection of no more than 0.2 classes (on average) that make the deterministic E-DEIM_{coh} an appealing alternative in this scenario.

The classes missed by E-DEIM_{coh} correspond to images of the letters B, C, D, I, O, R, and V; while it is unclear why these particular classes are missed, it is possible that not all

classes are detected due to the influence of the image perturbations and the effects they may have on making two images of the same letter appear as two separate classes. Like the MIT-BIH data set, we again expect to see the smaller amounts of dimension reduction for the E-DEIM implementations. For this particular data set, however, the extension of DEIM is clearly needed as there is no way for standard DEIM to identify all 26 classes in the data set from 10, 000 observations, even with the maximum amount of data reduction allowed.

With the ability to restart DEIM multiple times in E-DEIM, a topic of interest for future studies on this data set is the implementation of extended DEIM with the selection of more than $2k$ observations; seeing as we are able to identify 19 of the 26 classes in selecting only 32 observations with DEIM$_{coh}$, the question remains as to how many observations should be selected in order to detect all 26 classes. Without relying on a user-defined maximum number of indices, β, a subsequent question arises: What is an appropriate criteria to determine how many restarted DEIM iterations should be implemented to detect all of the relevant classes in the data set? The answer to this question remains an important area of interest for future work.

5.3 Discussion

In both the MIT-BIH Arrhythmia and Letter Recognition Sets, we see that the extension of DEIM with the inclusion of the coherence-computed $\mathbf{r}_2$ residual consistently outperforms standard DEIM for rank-k SVD approximations with $k > 1$. Even without the coherence means of maintaining "memory" of the previously selected indices, the extended versions of DEIM are indeed able to detect additional classes within the data upon restarting. Our experiments also indicate that DEIM$_S$ performs comparably to (or outperforms) k-medoid clustering and leverage scores in the selection of the same number of indices. In allowing for the selection of additional indices, E-DEIM$_{coh}$ consistently performs better than leverage scores, and is at least similar in performance, if not better, when compared to k-medoids and DEIM$_{rand}$, where the average performance of k-medoids and DEIM$_{rand}$ potentially depends on the uniformity of the class distributions throughout the data.

In carrying out these particular experiments, we are leveraging a labeled portion of the data set to select our parameters in DEIM and E-DEIM. While this may be recommended when such labels are available simply to gain a sense of some of the general data properties, we are not necessarily assuming that the same classes are present in both training and test sets; in fact, in our experiment set up, we know that there are actually more clinically-defined classes in the training set than in the test set for the MIT-BIH Arrhythmia data. Hence, even in tuning these parameters, we do not assume a fixed number of classes since a certain ratio of singular value decay may be achieved by different values of k. That being said, when labeled data is not available to identify a θ-value that maximizes class detection in the training set, there are a variety of ways to select k (or θ). For example, one option is to look for a "knee" in the plot of data matrix singular values (as is sometimes done with PCA). In the presence of computational constraints, another option is to simply select a feasible rank for the SVD. Or, if it is known that the number of classes likely exceeds the data matrix rank—the very scenario that inspired the development of this method (as in the letter data set)—one can simply take the full SVD of the data matrix and forgo selecting θ altogether. For both experiments conducted here, the selected values of θ result in rank-k SVD approximations with k often near or equal to the number time samples/features. The k-values are typically just shy of the number of samples per beat for the MIT-BIH Arrhythmia ECG data, allowing the number of selected beats to certainly exceed the

number of expected clinically-defined classes; perhaps as expected, k is equal to the number of features for the Letter Recognition Data Set with the selection of $2k$ observations only exceeding the number of expected classes by six observations. In addition to the previously described scenarios, even if the number of classes is known a priori in a labeled data set, an objective of applying E-DEIM can still be to identify a representative member (or members) of each known class, just as one could do in clustering the data until it is possible to identify representatives of each class.

As noted previously, careful tuning of τ is not really of great significance in our experiments as its primary role at this point is simply to avoid a singular $\hat{\mathbf{P}}_j^T \hat{\mathbf{V}}_{t_j}$ in forming the jth projection in extending DEIM. However, the incorporation of τ holds potential in defining stopping criteria for restarting DEIM multiple times because, if large enough, it can have an influence over which columns of $\hat{\mathbf{V}}$ are or are not included in forming the projection, $\hat{\mathcal{P}}$, during restarted DEIM; this is an area for future work.

Along with the development of stopping criteria for restarting DEIM in carrying out an extension, the theoretical implications of selecting additional indices through the E-DEIM methods presented here need additional attention. While we have presented slight generalizations of previous theoretical results in the application of E-DEIM to the CUR decomposition, the impact of selecting additional rows/columns through the inclusion of the "memory" residual, $\mathbf{r}_2$—in particular using coherence as opposed to the ℓ_1 distance—is not yet well understood despite observing positive results in practice. In addition, we reiterate that the error bounds discussed here certainly have their place in evaluating matrix approximations via subset selection, but the direct translation of these bounds to performance in the class identification setting is not obvious and remains a topic for future investigation.

We also note that while E-DEIM adds some computational cost to the standard DEIM algorithm, all of the DEIM/E-DEIM methods presented here, along with index selection through leverage scores, have a computational cost that is dominated by the formation of the SVD ($O(mnk)$). This is in comparison to the lower cost of the particular k-medoids algorithm implemented here ($O(mk)$), where a typical PAM implementation ($O(k(m-k)^2)$) would be more cost prohibitive than computing the SVD. However, the use of k-medoids leads to nondeterministic results with lingering questions about identifying an appropriate representative subset if the clustering is to be performed multiple times and in a truly unsupervised manner.

Having evaluated the performance of DEIM and E-DEIM in comparison with other methods applied to labeled data sets, we are able to observe that extending DEIM in a deterministic way is indeed a viable option in class-identification tasks—especially when the rank of the data matrix is not great enough for the detection of all classes. In particular, our results from the two different experiments discussed here support the use of E-DEIM$_{coh}$ in selecting additional indices. With room for further developing and understanding E-DEIM-type methods, the encouraging results presented here still suggest that the inclusion of such methods in completing machine learning tasks merits consideration.

6 Conclusion

In this work, we have presented a novel extension of the DEIM index-selection algorithm with the primary purpose of identifying additional data points of interest in the scenario that the number of classes present in the data set exceeds the matrix rank. As presented here, we are also able to study some of the implications of using E-DEIM to construct

the CUR matrix decomposition. In applying this algorithm to real data sets, we see that the extension does indeed allow for the identification of additional classes not detected with standard DEIM alone; the greatest improvement over standard DEIM is seen in the extension of DEIM with a "memory" residual defined in terms of the coherence between those rows/columns selected and those rows/columns left unselected via standard DEIM. The use of E-DEIM in class identification is further supported through the comparison of the proposed approach(es) with the more familiar k-medoids clustering algorithm and subset selection through statistical leverage scores. In our experiments, the deterministic E-DEIM$_{coh}$ outperforms or is at least comparable to the other methods in identifying class representatives.

With questions remaining regarding the number of extensions—or DEIM restarts—to include in the general algorithm presented here, this is an area of interest for future study along with a more theoretical analysis of E-DEIM in the class-identification setting. We also note that extensions of DEIM or DEIM-related algorithms can take on many forms, some arising in different fields and some yet to be developed. Given the success of our proposed E-DEIM algorithm in the experiments presented here, we plan to pursue this research area further for the purposes of class detection.

Appendix 1: Theoretical implications

While the measure of accuracy may be different in the theoretical and class-detection settings, we include these more theoretical results for completeness as extensions of DEIM may find use in other settings.

In extending DEIM to select $\hat{k}$ as opposed to k indices, we can also extend the related theoretical results. Where the results presented by Sorensen and Embree (2016) make use of the invertibility of $\mathbf{P}^T\mathbf{V}$ in standard DEIM, as mentioned above, we now only have that, for our extended DEIM construction of $\mathbf{P} \in \mathbb{R}^{m\times\hat{k}}$, $\mathbf{P}^T\mathbf{V}$ has full column rank. In the subsections that follow, we present the theoretical results for bounding the extended DEIM projection error and the subsequent results relating to the extended CUR factorization.

Extended DEIM projection error bound

The proof of the extended DEIM projection error for $\mathcal{P}$ defined as in Eq. (1) closely follows that presented by Sorensen and Embree (2016).

Lemma 2 *Let* $\mathbf{A}$ *be any matrix in* $\mathbb{R}^{m\times n}$, *and let* $\mathbf{V} \in \mathbb{R}^{m\times k}$ *be such that* $\mathbf{V}^T\mathbf{V} = \mathbf{I}$. *For* $\mathbf{P} = \mathbf{I}(:, \mathbf{p}) \in \mathbb{R}^{m\times\hat{k}}$, *assume* $\mathbf{P}^T\mathbf{V}$ *has full column rank. Let* $\mathcal{P} = \mathbf{V}(\mathbf{P}^T\mathbf{V})^\dagger\mathbf{P}^T$, *where* $(\mathbf{P}^T\mathbf{V})^\dagger = [(\mathbf{P}^T\mathbf{V})^T(\mathbf{P}^T\mathbf{V})]^{-1}(\mathbf{P}^T\mathbf{V})^T$. *Then, with* $\|\cdot\|$ *denoting the induced matrix* 2*-norm,*

$$\|\mathbf{A} - \mathcal{P}\mathbf{A}\| \leq \|(\mathbf{P}^T\mathbf{V})^\dagger\|\|(\mathbf{I} - \mathbf{V}\mathbf{V}^T)\mathbf{A}\|. \tag{6}$$

Furthermore, suppose $\mathbf{V}$ *contains the first k left singular vectors of* $\mathbf{A}$; *that is*, $\mathbf{V}$ *satisfies the rank-k approximation* $\mathbf{A} \approx \mathbf{V}\mathbf{S}\mathbf{W}^T$, *where* $\mathbf{V}^T\mathbf{V} = \mathbf{W}^T\mathbf{W} = \mathbf{I}$. *Then*

$$\|\mathbf{A} - \mathcal{P}\mathbf{A}\| \leq \|(\mathbf{P}^T\mathbf{V})^\dagger\|\sigma_{k+1}. \tag{7}$$

Proof Notice that

$$\mathcal{P}\mathbf{V} = \mathbf{V}[(\mathbf{P}^T\mathbf{V})^T(\mathbf{P}^T\mathbf{V})]^{-1}(\mathbf{P}^T\mathbf{V})^T\mathbf{P}^T\mathbf{V} = \mathbf{V},$$

which implies $(\mathbf{I} - \mathcal{P})\mathbf{V} = 0$. Then

$$\begin{aligned} ||\mathbf{A} - \mathcal{P}\mathbf{A}|| &= ||(\mathbf{I} - \mathcal{P})\mathbf{A}|| \\ &= ||(\mathbf{I} - \mathcal{P})(\mathbf{I} - \mathbf{V}\mathbf{V}^T)\mathbf{A}|| \\ &\leq \|\mathbf{I} - \mathcal{P}\|\|(\mathbf{I} - \mathbf{V}\mathbf{V}^T)\mathbf{A}\|. \end{aligned}$$

Using the fact that $\|\mathbf{I} - \mathcal{P}\| = \|\mathcal{P}\| = \|(\mathbf{P}^T\mathbf{V})^\dagger\|$ for $\mathcal{P} \neq \mathbf{I}$ or $\mathbf{0}$, the first result given by Eq. (6) holds. As suggested by Sorensen and Embree (2016), for proof that $\|\mathbf{I} - \mathcal{P}\| = \|\mathcal{P}\|$, see the work by Szyld (2006). The second equality, $\|\mathcal{P}\| = \|(\mathbf{P}^T\mathbf{V})^\dagger\|$, holds with $\|\cdot\|$ being the induced matrix 2-norm given that $\mathbf{P}$ and $\mathbf{V}$ both have orthonormal columns. If $\mathbf{V}$ contains the first k singular vectors of $\mathbf{A}$, then

$$\begin{aligned} \|\mathbf{A} - \mathcal{P}\mathbf{A}\| &\leq \|(\mathbf{P}^T\mathbf{V})^\dagger\|\|(\mathbf{I} - \mathbf{V}\mathbf{V}^T)\mathbf{A}\| \\ &= \|(\mathbf{P}^T\mathbf{V})^\dagger\|\|\mathbf{A} - \mathbf{V}\mathbf{S}\mathbf{W}^T\| \\ &= \|(\mathbf{P}^T\mathbf{V})^\dagger\|\sigma_{k+1}. \end{aligned}$$

Hence, the second result given by Eq. 7 holds. □

Having proved Lemma 2 for the row-selecting projector, $\mathcal{P}$, we note that a similar result holds for the column-selecting projector $\mathcal{Q} = \mathbf{Q}(\mathbf{W}^T\mathbf{Q})^\dagger\mathbf{W}^T$, where $(\mathbf{W}^T\mathbf{Q})^\dagger = (\mathbf{W}^T\mathbf{Q})^T[(\mathbf{W}^T\mathbf{Q})(\mathbf{W}^T\mathbf{Q})^T]^{-1}$. Specifically, for $\mathbf{W} \in \mathbb{R}^{n\times k}$ containing the first k right singular vectors of $\mathbf{A}$, and for $\mathbf{Q} = \mathbf{I}(:, \mathbf{q}) \in \mathbb{R}^{n\times\hat{k}}$ such that $\mathbf{W}^T\mathbf{Q}$ has full rank, we have the following projection error bound:

$$\|\mathbf{A} - \mathbf{A}\mathcal{Q}\| \leq \|(\mathbf{W}^T\mathbf{Q})^\dagger\|\sigma_{k+1}.$$

Theoretical implications for the CUR factorization

To prove an error bound for the CUR factorization with $\kappa > k$, $\mathbf{U}_1 = \mathbf{C}_1^\dagger\mathbf{A}\mathbf{R}^\dagger$, and $\mathbf{U}_2 = \mathbf{0}$ as described in Sect. 3.3, we will make use of the E-DEIM projection error that follows from Lemma 2 for the projection $\mathcal{Q}_1 = \mathbf{Q}_1(\mathbf{W}^T\mathbf{Q}_1)^\dagger\mathbf{W}^T$, where $\mathbf{Q}_1 = \mathbf{I}(:, \mathbf{q}_1)$ is such that $\mathbf{A}\mathbf{Q}_1 = \mathbf{C}_1$. Before proving this result, for completeness we first prove a slight generalization of Sorensen's and Embree's Lemma 4.2 (2016), closely following their proof technique.

Lemma 3 *Let $\mathbf{A} \approx \mathbf{V}\mathbf{S}\mathbf{W}^T$ be the rank-k SVD of $\mathbf{A}$ with $k < \min\{m, n\}$.*

Suppose that for $\hat{k} \geq k$, $\mathbf{p} \in \mathbb{R}^{\hat{k}}$ and $\mathbf{q} \in \mathbb{R}^{\hat{k}}$ are such that $\mathbf{C} = \mathbf{A}(:, \mathbf{q}) = \mathbf{A}\mathbf{Q}$ and $\mathbf{R} = \mathbf{A}(\mathbf{p}, :) = \mathbf{P}^T\mathbf{A}$, and $\|(\mathbf{P}^T\mathbf{V})^\dagger\|$ and $\|(\mathbf{W}^T\mathbf{Q})^\dagger\|$ are finite. Then

$$\|(\mathbf{I} - \mathbf{C}\mathbf{C}^\dagger)\mathbf{A}\| \leq \|(\mathbf{W}^T\mathbf{Q})^\dagger\|\sigma_{k+1} \quad \text{and} \quad \|\mathbf{A}(\mathbf{I} - \mathbf{R}^\dagger\mathbf{R})\| \leq \|(\mathbf{P}^T\mathbf{V})^\dagger\|\sigma_{k+1}. \tag{8}$$

Proof With $\mathbf{C}^\dagger = (\mathbf{C}\mathbf{C})^{-1}\mathbf{C}^T$ and $\mathbf{C} = \mathbf{A}\mathbf{Q}$, we see that

$$\mathbf{C}\mathbf{C}^\dagger\mathbf{A} = (\mathbf{A}\mathbf{Q})(\mathbf{Q}^T\mathbf{A}^T\mathbf{A}\mathbf{Q})^{-1}\mathbf{Q}^T\mathbf{A}^T\mathbf{A}.$$

Setting $\boldsymbol{\Phi} = \mathbf{Q}(\mathbf{Q}^T\mathbf{A}^T\mathbf{A}\mathbf{Q})^{-1}\mathbf{Q}^T\mathbf{A}^T\mathbf{A}$, it follows that $\mathbf{C}\mathbf{C}^\dagger\mathbf{A} = \mathbf{A}\boldsymbol{\Phi}$ and

$$(\mathbf{I} - \mathbf{C}\mathbf{C}^{\dagger})\mathbf{A} = \mathbf{A}(\mathbf{I} - \boldsymbol{\Phi}).$$

Notice that since $\boldsymbol{\Phi}$ is a projection onto the range of $\mathbf{Q}$, $\boldsymbol{\Phi}\mathbf{Q} = \mathbf{Q}$ and

$$\boldsymbol{\Phi}\mathcal{Q} = \boldsymbol{\Phi}\mathbf{Q}(\mathbf{W}^T\mathbf{Q})^{\dagger}\mathbf{W}^T = \mathbf{Q}(\mathbf{W}^T\mathbf{Q})^{\dagger}\mathbf{W}^T = \mathcal{Q}.$$

Then

$$\mathbf{A}(\mathbf{I} - \boldsymbol{\Phi}) = \mathbf{A}(\mathbf{I} - \boldsymbol{\Phi})(\mathbf{I} - \mathcal{Q}) = (\mathbf{I} - \mathbf{C}\mathbf{C}^{\dagger})\mathbf{A}(\mathbf{I} - \mathcal{Q})$$

so that $(\mathbf{I} - \mathbf{C}\mathbf{C}^{\dagger})\mathbf{A} = (\mathbf{I} - \mathbf{C}\mathbf{C}^{\dagger})\mathbf{A}(\mathbf{I} - \mathcal{Q})$. It follows, then that

$$\begin{aligned}\|(\mathbf{I} - \mathbf{C}\mathbf{C}^{\dagger})\mathbf{A}\| &= \|(\mathbf{I} - \mathbf{C}\mathbf{C}^{\dagger})\mathbf{A}(\mathbf{I} - \mathcal{Q})\| \\ &\leq \|\mathbf{I} - \mathbf{C}\mathbf{C}^{\dagger}\|\|\mathbf{A}(\mathbf{I} - \mathcal{Q})\| \\ &\leq \|(\mathbf{W}^T\mathbf{Q})^{\dagger}\|\sigma_{k+1}\end{aligned}$$

by Lemma 2 with $k < \min\{n, m\}$ and using the fact that $\mathbf{I} - \mathbf{C}\mathbf{C}^{\dagger}$ is an orthogonal projector with $\|\mathbf{I} - \mathbf{C}\mathbf{C}^{\dagger}\| = 1$. Hence the first inequality in (8) holds.

The proof of the second inequality in (8) follows a similar pattern. Defining $\boldsymbol{\Psi} = \mathbf{A}\mathbf{A}^T\mathbf{P}((\mathbf{P}^T\mathbf{A})(\mathbf{P}^T\mathbf{A})^T)^{-1}\mathbf{P}^T$, we see that

$$\begin{aligned}\mathbf{A}(\mathbf{I} - \mathbf{R}^{\dagger}\mathbf{R}) &= \mathbf{A} - \mathbf{A}\mathbf{A}^T\mathbf{P}((\mathbf{P}^T\mathbf{A})(\mathbf{P}^T\mathbf{A})^T)^{-1}\mathbf{P}^T\mathbf{A} \\ &= \mathbf{A} - \boldsymbol{\Psi}\mathbf{A} \\ &= (\mathbf{I} - \boldsymbol{\Psi})\mathbf{A} \\ &= (\mathbf{I} - \mathcal{P})(\mathbf{I} - \boldsymbol{\Psi})\mathbf{A} \\ &= (\mathbf{I} - \mathcal{P})\mathbf{A}(\mathbf{I} - \mathbf{R}^{\dagger}\mathbf{R}),\end{aligned}$$

where we recall that $\mathcal{P} = \mathbf{V}(\mathbf{P}^T\mathbf{V})^{\dagger}\mathbf{P}^T$. Then, it follows that

$$\begin{aligned}\|\mathbf{A}(\mathbf{I} - \mathbf{R}^{\dagger}\mathbf{R})\| &= \|(\mathbf{I} - \mathcal{P})\mathbf{A}(\mathbf{I} - \mathbf{R}^{\dagger}\mathbf{R})\| \\ &\leq \|(\mathbf{I} - \mathcal{P})\mathbf{A}\|\|(\mathbf{I} - \mathbf{R}^{\dagger}\mathbf{R})\| \\ &\leq \|(\mathbf{P}^T\mathbf{V})^{\dagger}\|\sigma_{k+1}.\end{aligned}$$

□

This result allows us to prove a bound on the CUR approximation error $\mathbf{A} \approx \mathbf{C}_1\mathbf{U}_1\mathbf{R}$ with $\mathbf{C}_1 \in \mathbb{R}^{m\times\kappa}$, $\mathbf{U}_1 \in \mathbb{R}^{\kappa\times k}$, and $\mathbf{R} \in \mathbb{R}^{k\times n}$ for $\kappa > k$ as described above. We first restate Theorem 1, and once again, we closely follow the technique presented by Sorensen and Embree (2016), which closely follows a technique by Mahoney and Drineas (2009).

Theorem 1 *Suppose* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *and* $k \leq rank(A)$ *with* $1 \leq k < \kappa \leq \min\{m, n\}$. *Let* $\mathbf{C}_1 = \mathbf{A}(:, \mathbf{q}_1) \in \mathbb{R}^{m\times\kappa}$, $\mathbf{R} = \mathbf{A}(\mathbf{p}, :) \in \mathbb{R}^{k\times n}$ *and* $\mathbf{U}_1 = \mathbf{C}_1^{\dagger}\mathbf{A}\mathbf{R}^{\dagger}$. *Then*

$$\|\mathrm{A} - \mathrm{C}_1\mathrm{U}_1\mathrm{R}\| \leq (\|(\mathrm{P}^T\mathrm{V})^{-1}\| + \|(\mathrm{W}^T\mathrm{Q}_1)^{\dagger}\|)\sigma_{k+1}. \tag{9}$$

Proof Since $\mathbf{U}_1 = \mathbf{C}_1^{\dagger}\mathbf{A}\mathbf{R}^{\dagger}$,

$$\mathbf{A} - \mathbf{C}_1\mathbf{U}_1\mathbf{R} = \mathbf{A} - \mathbf{C}_1\mathbf{C}_1^{\dagger}\mathbf{A}\mathbf{R}^{\dagger}\mathbf{R} = (\mathbf{I} - \mathbf{C}_1\mathbf{C}_1^{\dagger})\mathbf{A} + \mathbf{C}_1\mathbf{C}_1^{\dagger}\mathbf{A}(\mathbf{I} - \mathbf{R}^{\dagger}\mathbf{R})$$

Then

$$\begin{aligned}\|\mathbf{A}-\mathbf{C}_1\mathbf{U}_1\mathbf{R}\| &= \|(\mathbf{I}-\mathbf{C}_1\mathbf{C}_1^{\dagger})\mathbf{A}+\mathbf{C}_1\mathbf{C}_1^{\dagger}\mathbf{A}(\mathbf{I}-\mathbf{R}^{\dagger}\mathbf{R})\| \\ &\leq \|(\mathbf{I}-\mathbf{C}_1\mathbf{C}_1^{\dagger})\mathbf{A}\|+\|\mathbf{C}_1\mathbf{C}_1^{\dagger}\|\|\mathbf{A}(\mathbf{I}-\mathbf{R}^{\dagger}\mathbf{R})\| \\ &\leq \|(\mathbf{W}^T\mathbf{Q}_1)^{\dagger}\|\sigma_{k+1}+\|(\mathbf{P}^T\mathbf{V})^{-1}\|\sigma_{k+1}.\end{aligned}$$

This last line follows from Lemma 3 above and the related standard-DEIM result presented in Lemma 4.2 by Sorensen and Embree (2016). Hence we see the result in (9) holds.□

Appendix 2: MIT-BIH Arrhythmia Data Set training/test split

As described by De Chazal et al. (2004) in splitting the MIT-BIH Arrhythmia ECG data, the training set "DS1" contains files 101, 106, 108, 109, 112, 114, 115, 116, 118, 119, 122, 124, 201, 203, 205, 207, 208, 209, 215, 220, 223, 230.

The test set "DS2" contains files 100, 103, 105, 111, 113, 117, 121, 123, 200, 202, 210, 212, 213, 214, 219, 221, 222, 228, 231, 232, 233, 234. Note that the four files containing paced beats (102, 104, 107, and 217) are not included in the training or test sets (De Chazal et al., 2004).

Acknowledgements The authors would like to thank Danny Sorensen for his helpful comments throughout the development of this work. We also appreciate the suggestions and feedback offered by the editor and anonymous reviewers during the revision process. This research was primarily supported by a training fellowship from the Keck Center of the Gulf Coast Consortia, on the NLM Training Program in Biomedical Informatics (NLM Grant No. T15LM007093). Partial support came from the following: NSF DMS-0739420, NSF DMS-1312391, NSF DMS-1318348, NIH 1R56HL131574, and Award 16BGIA27490024 from the American Heart Association and The Children's Heart Foundation.

Compliance with ethical standards

Conflict of interest The only potential conflict to report is that Craig Rusin is a co-founder of Medical Informatics Corp. Medical Informatics Corp has no financial interest in this study. All other authors report no conflicts.

References

Candes, E. J., Eldar, Y. C., Needell, D., & Randall, P. (2011). Compressed sensing with coherent and redundant dictionaries. *Applied and Computational Harmonic Analysis, 31*(1), 59–73. https://doi.org/10.1016/j.acha.2010.10.002.

Chaturantabut, S., & Sorensen, D. C. (2010). Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing, 32*(5), 2737–2764. https://doi.org/10.1137/090766498.

De Chazal, P., O'Dwyer, M., & Reilly, R. B. (2004). Automatic classification of heartbeats using ECG morphology and heartbeat interval features. *IEEE Transactions on Biomedical Engineering, 51*(7), 1196–1206. https://doi.org/10.1109/TBME.2004.827359.

Donoho, D. L., & Huo, X. (2001). Uncertainty principles and ideal atomic decomposition. *IEEE Transactions on Information Theory, 47*(7), 2845–2862. https://doi.org/10.1109/18.959265.

Drineas, P., Mahoney, M. W., & Muthukrishnan, S. (2008). Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications, 30*(2), 844–881. https://doi.org/10.1137/07070471X.

Drmač, Z., & Gugercin, S. (2016). A new selection operator for the discrete empirical interpolation method-improved a priori error bound and extensions. *SIAM Journal on Scientific Computing, 38*(2), A631–A648.

Dua, D., & Taniskidou, E. K. (2017). UCI machine learning repository. http://archive.ics.uci.edu/ml.
Frey, P. W., & Slate, D. J. (1991). Letter recognition using Holland-style adaptive classifiers. *Machine Learning, 6*(2), 161–182. https://doi.org/10.1007/BF00114162.
Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., et al. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation, 101*(23), e215–e220. https://doi.org/10.1161/01.CIR.101.23.e215.
Goreinov, S. A., Tyrtyshnikov, E. E., & Zamarashkin, N. L. (1997). A theory of pseudoskeleton approximations. *Linear Algebra and Its Applications, 261*(1), 1–21.
Hendryx, E. P. (2018). *Subset selection and feature identification in the electrocardiogram*. Ph.D. thesis, Rice University.
Hendryx, E. P., Rivière, B. M., Sorensen, D. C., & Rusin, C. G. (2018). Finding representative electrocardiogram beat morphologies with CUR. *Journal of Biomedical Informatics, 77,* 97–110. https://doi.org/10.1016/j.jbi.2017.12.003.
Mahoney, M. W., & Drineas, P. (2009). CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences, 106*(3), 697–702. https://doi.org/10.1073/pnas.0803205106.
Manohar, K., Brunton, B. W., Kutz, J. N., & Brunton, S. L. (2018). Data-driven sparse sensor placement for reconstruction: Demonstrating the benefits of exploiting known patterns. *IEEE Control Systems Magazine, 38*(3), 63–86.
MATLAB. (2020). MathWorks Documentation: Kmedoids. MATLAB_R2020a edn. https://www.mathworks.com/help/stats/kmedoids.html.
Moody, G. B., & Mark, R. G. (2001). The impact of the MIT-BIH arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine, 20*(3), 45–50. https://doi.org/10.1109/51.932724.
Park, H. S., & Jun, C. H. (2009). A simple and fast algorithm for K-medoids clustering. *Expert Systems with Applications, 36*(2), 3336–3341.
Peherstorfer, B., Drmač, Z., & Gugercin, S. (2018). Stabilizing discrete empirical interpolation via randomized and deterministic oversampling. arXiv preprint arXiv:180810473.
Sorensen, D. C., & Embree, M. (2016). A DEIM induced CUR factorization. *SIAM Journal on Scientific Computing, 38*(3), A1454–A1482. https://doi.org/10.1137/140978430.
Stewart, G. (1999). Four algorithms for the efficient computation of truncated pivoted QR approximations to a sparse matrix. *Numerische Mathematik, 83*(2), 313–323. https://doi.org/10.1007/s002110050451.
Szyld, D. B. (2006). The many proofs of an identity on the norm of oblique projections. *Numerical Algorithms, 42*(3–4), 309–323. https://doi.org/10.1007/s11075-006-9046-2.
Zhou, Y. B. (2012). *Model reduction for nonlinear dynamical systems with parametric uncertainties*. Ph.D. thesis, Massachusetts Institute of Technology.
Zimmermann, R., & Willcox, K. (2016). An accelerated greedy missing point estimation procedure. *SIAM Journal on Scientific Computing, 38*(5), A2827–A2850.

Authors and Affiliations

Emily P. Hendryx[1] · Béatrice M. Rivière[2] · Craig G. Rusin[3]

Béatrice M. Rivière
riviere@rice.edu

Craig G. Rusin
crusin@bcm.edu

[1] Department of Mathematics and Statistics, University of Central Oklahoma, Edmond, OK 73034-5207, USA

[2] Department of Computational and Applied Mathematics, Rice University, Houston, TX 77005-1892, USA

[3] Department of Pediatric Cardiology, Baylor College of Medicine, Houston, TX 77030-1892, USA