Scalable Unsupervised Feature Selection with Reconstruction Error Guarantees via QMR Decomposition

Ciwan Ceylan

KTH Royal Institute of Technology
Division of Robotics, Perception, and Learning

SEB Group

Abstract

Unsupervised feature selection (UFS) methods are valuable for eliminating redundant features without using class labels, but they often struggle with large datasets. To overcome this, we present QMR-FS, a scalable, greedy forward filtering approach that selects linearly independent features while ensuring any excluded features can be reconstructed within a specified tolerance. QMR-FS is based on QMR matrix decomposition, an extension of QR decomposition, resulting in linear complexity relative to the number of samples. This facilitates parallelized computation on both CPU and GPU, and our model implementation runs in a matter of seconds on datasets with up to 1 billion elements, while offering classification and clustering performance comparable to other UFS methods.

Problem: The UFS task is to select a subset of the columns of a data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, while preserving or improving downstream task accuracy. We desire a method which scales to large *n*.

QMR-FS is a greedy approach based on linear independence.

$$\mathbf{x}_j = \sum_{k=1}^{j-1} a_k \mathbf{x}_k + b\mathbf{1}_n$$

Iterating over the columns in **X** from left to right, we drop columns which can be expressed as a linear combination of previously seen columns. Specifically, column *j* is dropped if there exists coefficients a_k and *b* that fulfills the formula above.

Although this could be done by performing *d* least-squares fits, it would be inefficient, with a time complexity of $O(nd^3)$.

Instead, QMR-FS archives time complexity $O(nd^2)$ by utilizing the following lemma, which we prove in our paper.

LEMMA 2.1. *Let* $\mathbf{X} \in \mathbb{R}^{n \times d}$ *and* $\mathbf{R}_{ref} \in \mathbb{R}^{d \times d}$ *satisfy* $\mathbf{X} = \mathbf{U}\mathbf{R}_{ref}$ *and* $\mathbf{R}_{ref} = \mathbf{U}^{\dagger}\mathbf{X}$, *where* $\mathbf{R}_{ref}$ *is in row echelon form (REF), and* $\mathbf{U} \in \mathbb{R}^{n \times d}$ *has full rank and the left-side inverse* $\mathbf{U}^{\dagger} \in \mathbb{R}^{d \times n}$. *Then* $\mathbf{x}_j \in \text{span}(\mathbf{x}_1, \ldots, \mathbf{x}_{j-1})$ ***iff*** $\mathbf{r}_j \in \text{span}(\mathbf{r}_1, \ldots, \mathbf{r}_{j-1})$ *for any* $j > 1$ *where* $\mathbf{x}_j \in \mathbb{R}^n$ *is the* j*th column in* $\mathbf{X}$, *and* $\mathbf{r}_j \in \mathbb{R}^d$ *is the* j*th column in* $\mathbf{R}_{ref}$.

The lemma requires a matrix decomposition $\mathbf{X}=\mathbf{U}\mathbf{R}_{ref}$ where **U** has a left-side inverse, and $\mathbf{R}_{ref}$ is in row echelon form. The QR decomposition is a good start, but insufficient since **R** is only upper triangular. Here is an example showcasing this limitation.

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 2 & 1 & 1 \\ 1 & 0 & 0 & 1 & 2 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{Q} = \begin{bmatrix} -0.45 & 0.89 & 0. & 0. & 0. \\ -0.45 & -0.22 & 0.87 & 0. & 0. \\ -0.45 & -0.22 & -0.29 & 0.82 & 0. \\ -0.45 & -0.22 & -0.29 & -0.41 & -0.71 \\ -0.45 & -0.22 & -0.29 & -0.41 & 0.71 \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} -2.24 & -0.45 & -0.89 & -1.34 & -1.79 \\ 0. & 0.89 & 1.79 & 0.45 & 0.22 \\ 0. & 0. & 0. & 0.58 & 1.44 \\ 0. & 0. & 0. & 0.82 & 0.82 \\ 0. & 0. & 0. & 0. & 0. \end{bmatrix}$$

As **R** is not in row echelon form, it is not clear that columns 4 and 5 are both linearly independent. Therefore, **R** must be further decomposed using Gaussian elimination. This results in the QMR decomposition: $\mathbf{X}=\mathbf{Q}\mathbf{M}\mathbf{R}_{ref}$:

$$\mathbf{M} = \begin{bmatrix} 1. & 0. & 0. & 0. & 0. \\ 0. & 1. & 0. & 0. & 0. \\ 0. & 0. & 1. & 0. & 0. \\ 0. & 0. & 1.41 & 1. & 0. \\ 0. & 0. & 0. & 0. & 1. \end{bmatrix} \quad \mathbf{R}_{\text{ref}} = \begin{bmatrix} -2.2 & -0.45 & -0.89 & -1.34 & -1.79 \\ 0. & 0.89 & 1.79 & 0.45 & 0.22 \\ 0. & 0. & 0. & 0.58 & 1.44 \\ 0. & 0. & 0. & 0. & -1.23 \\ 0. & 0. & 0. & 0. & 0. \end{bmatrix}$$

Gaussian elimination is invertible, meaning that **QM** has the left inverse matrix $\mathbf{M}^{-1}\mathbf{Q}^{\mathsf{T}}$. Thus, by the lemma, the linearly independent columns in **X** are the same as those in $\mathbf{R}_{ref}$, which are identified by the pivot elements. In the example, column 3 can be removed.

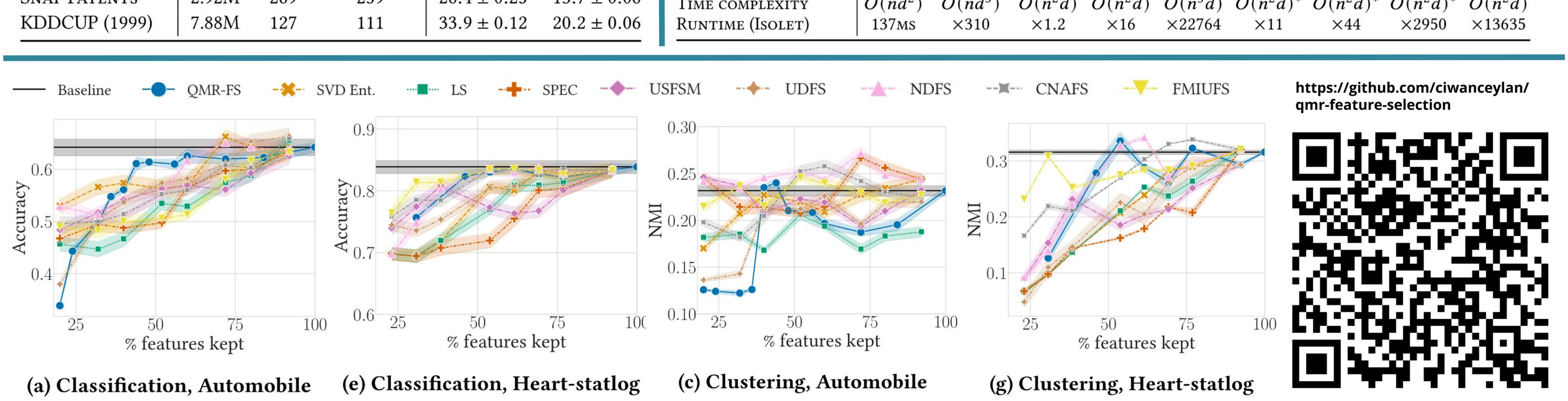
Thresholding: In practice, exact linear independence can be too strict, leading to more features being retained than desired. To address this, QMR-FS removes features with small reconstruction errors. During the Gaussian elimination, we can efficiently compute an upper bound on the reconstruction error without expensive least-square fitting. See our paper for more details.

Table 1: QMR-FS runtimes in seconds on four large datasets using threshold value 0.1, resulting in d_{fs} selected features.

	Num. instances and dims.			Runtime (s)	
Dataset	*n*	*d*	d_{fs}	CPU	GPU
US Census (1990)	2.46M	68	66	4.16 ± 0.16	2.75 ± 0.11
GitHub MUSAE	37.7K	4006	3799	28.5 ± 0.18	13.9 ± 0.45
Snap Patents	2.92M	269	259	26.4 ± 0.25	13.7 ± 0.06
KDDCUP (1999)	7.88M	127	111	33.9 ± 0.12	20.2 ± 0.06

Table 2: Summary of benchmark results using 40% and 60% kept features. The average ranks and standard deviations for classification and clustering are computed over the 6 datasets. The time complexities are simplified under the assumption $n \geq d$, and * indicates methods which are iterated until convergence. Relative runtimes are displayed for Isolet, the largest dataset the baseline UFS methods scale to.

	QMR-FS	SVD Ent.	LS	SPEC	USFSM	UDFS	NDFS	CNAFS	FMIUFS
Clsif. avg. rank (40%)	2.8 ± 1.8	5.5 ± 2.8	5.8 ± 2.6	8.3 ± 0.5	3.0 ± 2.2	5.5 ± 1.8	4.3 ± 1.9	5.0 ± 1.4	3.7 ± 2.6
Clsif. avg. rank (60%)	2.5 ± 1.6	5.0 ± 1.8	5.7 ± 2.1	6.5 ± 2.6	5.8 ± 2.8	4.7 ± 2.8	3.2 ± 1.7	5.3 ± 3.1	5.2 ± 3.1
Clstr. avg. rank (40%)	3.0 ± 2.2	6.7 ± 2.1	6.5 ± 2.6	7.0 ± 1.7	3.7 ± 2.1	6.0 ± 2.8	3.3 ± 2.3	5.5 ± 1.6	2.8 ± 1.3
Clstr. avg. rank (60%)	5.5 ± 2.9	4.3 ± 2.0	4.8 ± 2.4	7.2 ± 1.3	4.8 ± 2.5	5.5 ± 2.6	3.0 ± 2.7	4.2 ± 3.4	4.8 ± 2.3
Time complexity	$O(nd^2)$	$O(nd^3)$	$O(n^2d)$	$O(n^2d)$	$O(n^3d)$	$O(n^2d)^*$	$O(n^2d)^*$	$O(n^2d)^*$	$O(n^2d)$
Runtime (Isolet)	137ms	×310	×1.2	×16	×22764	×11	×44	×2950	×13635



(a) Classification, Automobile (e) Classification, Heart-statlog (c) Clustering, Automobile (g) Clustering, Heart-statlog

https://github.com/ciwanceylan/qmr-feature-selection

WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM