1 Supplemental Material

1.1 Theorems and Proofs

Theorem 1 *Let $\boldsymbol{x}^{t+1}$ be the new row added to $\boldsymbol{X}^t$ at iteration $t+1$. If the QR factorization $\boldsymbol{Q}^t_{\boldsymbol{X}}\boldsymbol{R}^t_{\boldsymbol{X}} = \boldsymbol{X}^t$ is known, then*

$$\boldsymbol{H}^{t+1} = \begin{bmatrix} 1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{H}^t \end{bmatrix} - \boldsymbol{v}\boldsymbol{v}' \tag{1}$$

where $\boldsymbol{v}$ is the last column of $\boldsymbol{Q}^{t+1}_{\boldsymbol{X}}$.

Proof: We start by finding the new QR factorization of $\boldsymbol{X}^{t+1}$.

$$\boldsymbol{X}^{t+1} = \begin{bmatrix} \boldsymbol{x}^{t+1} \\ \boldsymbol{X}^t \end{bmatrix} \tag{2}$$

$\boldsymbol{Q}^t_{\boldsymbol{X}}$ is an orthonormal basis for $\boldsymbol{X}^t$, meaning $\boldsymbol{Q}^{t\prime}_{\boldsymbol{X}}\boldsymbol{Q}^t_{\boldsymbol{X}} = \boldsymbol{I}$, and thus $\boldsymbol{Q}^{t\prime}_{\boldsymbol{X}}\boldsymbol{X}^t = \boldsymbol{R}^t_{\boldsymbol{X}}$. We can expand the size of $\boldsymbol{Q}^t_{\boldsymbol{X}}$ to account for the increased size of $\boldsymbol{X}^{t+1}$ while maintaining orthonormality, giving us

$$\begin{bmatrix} 1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}^{t\prime}_{\boldsymbol{X}} \end{bmatrix} \boldsymbol{X}^{t+1} = \begin{bmatrix} 1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}^{t\prime}_{\boldsymbol{X}} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^{t+1} \\ \boldsymbol{X}^t \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}^{t+1} \\ \boldsymbol{R}^t_{\boldsymbol{X}} \end{bmatrix} = \tilde{\boldsymbol{R}}_{\boldsymbol{X}} \tag{3}$$

In equation 3, we use a boldface zero ($\boldsymbol{0}$) to indicate the rest of the row or column of the matrix is filled with 0s.

Our goal is then to refactor $\tilde{\boldsymbol{R}}_{\boldsymbol{X}}$ into an upper triangular matrix while preserving the orthonormality of $\boldsymbol{Q}^{t+1}_{\boldsymbol{X}}$. Givens rotations are a common tool used to accomplish this refactoring efficiently in QR factorization algorithms. $\tilde{\boldsymbol{R}}_{\boldsymbol{X}}$ is upper Hessenberg, meaning we can quickly construct the upper triangular matrix $\boldsymbol{G}'_{\boldsymbol{X}}\tilde{\boldsymbol{R}}_{\boldsymbol{X}}$ using only p Givens rotations $\boldsymbol{G}_{\boldsymbol{X},1}\ldots\boldsymbol{G}_{\boldsymbol{X},p} = \boldsymbol{G}_{\boldsymbol{X}}$ (where $\boldsymbol{G}_{\boldsymbol{X},k}$ indicates the kth Givens rotation matrix). Note that $\boldsymbol{G}_{\boldsymbol{X}}\boldsymbol{G}'_{\boldsymbol{X}} = \boldsymbol{I}$. Thus we have

$$\boldsymbol{G}'_{\boldsymbol{X}} \begin{bmatrix} 1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}^{t\prime}_{\boldsymbol{X}} \end{bmatrix} \boldsymbol{X}^{t+1} = \boldsymbol{G}'_{\boldsymbol{X}} \begin{bmatrix} 1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}^{t\prime}_{\boldsymbol{X}} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^{t+1} \\ \boldsymbol{X}^t \end{bmatrix} \tag{4}$$

$$= \boldsymbol{G}'_{\boldsymbol{X}} \begin{bmatrix} \boldsymbol{x}^{t+1} \\ \boldsymbol{R}^t_{\boldsymbol{X}} \end{bmatrix} \tag{5}$$

$$= \boldsymbol{R}^{t+1}_{\boldsymbol{X}} \tag{6}$$

$$\boldsymbol{Q}^{t+1}_{\boldsymbol{X}} = \begin{bmatrix} 1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}^t_{\boldsymbol{X}} \end{bmatrix} \boldsymbol{G}_{\boldsymbol{X}} \tag{7}$$

However, in this form $\boldsymbol{Q}^{t+1}_{\boldsymbol{X}}$ is an $(n+1)\times(p+1)$ matrix, which has one more column than we need for an orthonormal basis of $\boldsymbol{X}^{t+1}$. $\boldsymbol{R}^{t+1}_{\boldsymbol{X}}$ has dimensions $(p+1)\times p$ and is almost a triangular matrix except its last row is $\boldsymbol{0}$. This means that the last column of $\boldsymbol{Q}^{t+1}_{\boldsymbol{X}}$, denoted as $\boldsymbol{v}$, is part of the left null space of $\boldsymbol{X}^{t+1}$ and contributes nothing to the reconstruction of $\boldsymbol{X}^{t+1}$. Thus we can safely remove this column from $\boldsymbol{Q}^{t+1}_{\boldsymbol{X}}$ and the last row of $\boldsymbol{R}^{t+1}_{\boldsymbol{X}}$ without changing the factorization.

We can represent this removal mathematically by multiplication by $\begin{bmatrix} \boldsymbol{I}_p \\ \boldsymbol{0} \end{bmatrix}$, where $\boldsymbol{I}_p$ is the $p\times p$ identity matrix.

Recall that $\boldsymbol{H} = \boldsymbol{Q}_{\boldsymbol{X}}\boldsymbol{Q}'_{\boldsymbol{X}}$ when $\boldsymbol{Q}_{\boldsymbol{X}}$ is an orthonormal basis for $\boldsymbol{X}$. Let us denote the last column of $\boldsymbol{Q}^{t+1}_{\boldsymbol{X}}$ as $\boldsymbol{v}$. Using the vector removal notation from above, we can rewrite the form of $\boldsymbol{H}^{t+1}$, where $\boldsymbol{0}_p$ is the $p\times p$ zero matrix:

$$\boldsymbol{H}^{t+1} = \boldsymbol{Q}^{t+1}_{\boldsymbol{X}} \begin{bmatrix} \boldsymbol{I}_p \\ \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_p & \boldsymbol{0} \end{bmatrix} \boldsymbol{Q}^{t+1\prime}_{\boldsymbol{X}} = \tag{8}$$

$$\boldsymbol{Q}^{t+1}_{\boldsymbol{X}} \left(\boldsymbol{I}_{p+1} - \begin{bmatrix} \boldsymbol{0}_p & \boldsymbol{0} \\ \boldsymbol{0} & 1 \end{bmatrix} \right) \boldsymbol{Q}^{t+1\prime}_{\boldsymbol{X}} = \tag{9}$$

$$\boldsymbol{Q}^{t+1}_{\boldsymbol{X}}\boldsymbol{Q}^{t+1\prime}_{\boldsymbol{X}} - \boldsymbol{Q}^{t+1}_{\boldsymbol{X}} \begin{bmatrix} \boldsymbol{0}_p & \boldsymbol{0} \\ \boldsymbol{0} & 1 \end{bmatrix} \boldsymbol{Q}^{t+1\prime}_{\boldsymbol{X}} = \tag{10}$$

$$\begin{bmatrix} 1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}^t_{\boldsymbol{X}} \end{bmatrix} \boldsymbol{G}_{\boldsymbol{X}}\boldsymbol{G}'_{\boldsymbol{X}} \begin{bmatrix} 1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}^{t\prime}_{\boldsymbol{X}} \end{bmatrix} - \boldsymbol{Q}^{t+1}_{\boldsymbol{X}} \begin{bmatrix} \boldsymbol{0} \\ 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{0} & 1 \end{bmatrix} \boldsymbol{Q}^{t+1\prime}_{\boldsymbol{X}} = \tag{11}$$

$$\begin{bmatrix} 1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}^t_{\boldsymbol{X}}\boldsymbol{Q}^{t\prime}_{\boldsymbol{X}} \end{bmatrix} - \boldsymbol{v}\boldsymbol{v}' = \tag{12}$$

$$\begin{bmatrix} 1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{H}^t \end{bmatrix} - \boldsymbol{v}\boldsymbol{v}' \tag{13}$$

□

Theorem 2 *If $\boldsymbol{Z}^t$ and $\boldsymbol{Q}^{t+1}_{\boldsymbol{X}}$ are known, then*

$$\boldsymbol{Z}^{t+1} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{Z}^t \end{bmatrix} + \boldsymbol{v}\boldsymbol{g}' \tag{14}$$

where $\boldsymbol{g} = [\tilde{\boldsymbol{x}}^{t+1}, \tilde{\boldsymbol{X}}^t]\boldsymbol{v}$ and $\boldsymbol{v}$ is the last column of $\boldsymbol{Q}^{t+1}_{\boldsymbol{X}}$.

Proof: From Theorem 1 we know how to express the form of $\boldsymbol{H}^{t+1}$. Let $\tilde{\boldsymbol{x}}^{t+1}$ be the row added to $\tilde{\boldsymbol{X}}^t$.

$$\boldsymbol{Z}^{t+1} = \tilde{\boldsymbol{X}}^{t+1} - \boldsymbol{H}^{t+1}\tilde{\boldsymbol{X}}^{t+1} = \tag{15}$$

$$\begin{bmatrix}\tilde{\boldsymbol{x}}^{t+1}\\ \tilde{\boldsymbol{X}}^t\end{bmatrix} - \left(\begin{bmatrix}1 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{H}^t\end{bmatrix} - \boldsymbol{v}\boldsymbol{v}'\right)\begin{bmatrix}\tilde{\boldsymbol{x}}^{t+1}\\ \tilde{\boldsymbol{X}}^t\end{bmatrix} = \tag{16}$$

$$\begin{bmatrix}\tilde{\boldsymbol{x}}^{t+1}\\ \tilde{\boldsymbol{X}}^t\end{bmatrix} - \begin{bmatrix}\tilde{\boldsymbol{x}}^{t+1}\\ \boldsymbol{H}^t\tilde{\boldsymbol{X}}^t\end{bmatrix} + \boldsymbol{v}\boldsymbol{v}'\begin{bmatrix}\tilde{\boldsymbol{x}}^{t+1}\\ \tilde{\boldsymbol{X}}^t\end{bmatrix} = \tag{17}$$

$$\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{Z}^t\end{bmatrix} + \boldsymbol{v}\boldsymbol{g}' \tag{18}$$

where $\boldsymbol{g} = [\tilde{\boldsymbol{x}}^{t+1}, \tilde{\boldsymbol{X}}^t]\boldsymbol{v}$. □

Theorem 3 *Assume the QR factorization* $\boldsymbol{Z}^t = \boldsymbol{Q}_{\boldsymbol{Z}}^t\boldsymbol{R}_{\boldsymbol{Z}}^t$ *is known. Two sets of* $O(p)$ *Givens rotations* $\boldsymbol{G}_{\boldsymbol{R}} = \boldsymbol{G}_{\boldsymbol{R},1}\ldots\boldsymbol{G}_{\boldsymbol{R},(p-1)}$ *and* $\boldsymbol{G}_{\boldsymbol{B}} = \boldsymbol{G}_{\boldsymbol{B},1}\ldots\boldsymbol{G}_{\boldsymbol{B},p}$ *can be constructed such that the factorization of* $\boldsymbol{Z}^{t+1}$ *is*

$$\boldsymbol{Z}^{t+1} = \boldsymbol{Q}_{\boldsymbol{Z}}^{t+1}\boldsymbol{R}_{\boldsymbol{Z}}^{t+1} \tag{19}$$

$$\boldsymbol{Q}_{\boldsymbol{Z}}^{t+1} = \boldsymbol{Q}^t\boldsymbol{G}_{\boldsymbol{R}}'\boldsymbol{G}_{\boldsymbol{B}}' \tag{20}$$

$$\boldsymbol{R}_{\boldsymbol{Z}}^{t+1} = \boldsymbol{G}_{\boldsymbol{B}}\boldsymbol{G}_{\boldsymbol{R}}\Big(\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{R}_{\boldsymbol{Z}}^t\end{bmatrix} + \boldsymbol{c}\boldsymbol{g}'\Big) \tag{21}$$

where $\boldsymbol{c} = \boldsymbol{Q}_{\boldsymbol{Z}}^{t\prime}\boldsymbol{v}$ *and* $\boldsymbol{g}$ *is as defined in Theorem 2.*

Proof: Given an existing decomposition $\boldsymbol{Z}^t = \boldsymbol{Q}_{\boldsymbol{Z}}^t\boldsymbol{R}_{\boldsymbol{Z}}^t$,

$$\begin{aligned}\boldsymbol{Z}^{t+1} &= \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{Z}^t\end{bmatrix} + \boldsymbol{v}\boldsymbol{g}'\\ &= \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{Q}_{\boldsymbol{Z}}^t\boldsymbol{R}_{\boldsymbol{Z}}^t\end{bmatrix} + \boldsymbol{v}\boldsymbol{g}'\\ &= \boldsymbol{Q}_{\boldsymbol{Z}}^t\Big(\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{R}_{\boldsymbol{Z}}^t\end{bmatrix} + \boldsymbol{c}\boldsymbol{g}'\Big)\end{aligned}$$

where $\boldsymbol{c} = \boldsymbol{Q}_{\boldsymbol{Z}}^{t\prime}\boldsymbol{v}$.

We can compute a set of $(p-1)$ Givens rotations $\boldsymbol{G}_{\boldsymbol{R}} = \boldsymbol{G}_{\boldsymbol{R},1}\ldots\boldsymbol{G}_{\boldsymbol{R},(p-1)}$ such that $\boldsymbol{G}_{\boldsymbol{R}}\boldsymbol{c} = ||\boldsymbol{c}||\boldsymbol{e_1}$ where $\boldsymbol{e_1}$ is the first unit basis vector. Then $\boldsymbol{G}_{\boldsymbol{R}}(\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{R}_{\boldsymbol{Z}}^t\end{bmatrix} + \boldsymbol{c}\boldsymbol{g}') = \begin{bmatrix}\boldsymbol{0}\\ \tilde{\boldsymbol{R}}\end{bmatrix} + ||\boldsymbol{c}||\boldsymbol{e}_1\boldsymbol{g}' = \boldsymbol{B}$. The matrix $\boldsymbol{B}$ is guaranteed to be upper Hessenberg; as before, we can construct p Givens rotations $\boldsymbol{G}_{\boldsymbol{B}} = \boldsymbol{G}_{\boldsymbol{B},1}\ldots\boldsymbol{G}_{\boldsymbol{B},p}$ such that $\boldsymbol{G}_{\boldsymbol{B}}\boldsymbol{B} = \boldsymbol{R}^{t+1}$ is upper triangular. It then follows that $\boldsymbol{Q}^{t+1} = \boldsymbol{Q}^t\boldsymbol{G}_{\boldsymbol{R}}'\boldsymbol{G}_{\boldsymbol{B}}'$. □

1.2 Reproduction of Results

Code to reproduce the results documented in this paper can be found at https://github.com/Qsnap/AIStats20. The repository contains matlab code for running Qsnap, TESS, SSS-Moods, and the non-optimized rank-test quantile snapshot scan algorithm. It contains scripts to exactly reproduce the experiments detailed in this paper. The simulation data from our experiments, as well as the code for generated the simulated data, are included, along with the subset of eBird and climate data used in the experiments.

1.3 Partial AUC Calculation

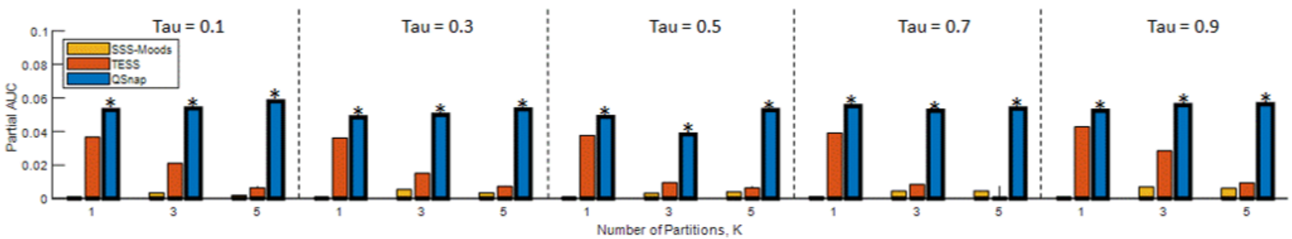
For our simulation experiments we evaluate each algorithm by computing the partial AUC of the TPR vs FPR graph, over the FPR range [0,0.2]. For each of the 30 datasets in each experiment setup, each algorithm reports the best region discovered, which we denote as $\boldsymbol{C}^*$. Any points in $\boldsymbol{C}^*$ that are generated from the shifted distribution are true positives (TP), while all other points in $\boldsymbol{C}^*$ are false positives (FP). Points outside $\boldsymbol{C}^*$ generated by the shifted distribution are false negatives (FN), while the other data points outside C^* are true negatives (TN).

For each dataset, we compute the TPR and FPR of the best region from each algorithm as $tpr = \frac{TP}{TP+FN}$ and $fpr = \frac{FP}{FP+TN}$. In our setup we must either accept the entire region or none of it. This means that TPR = 0 when FPR $<$ fpr, and TPR = tpr when FPR $\geq$ fpr. This produces a step graph for each algorithm on each dataset. We calculate the partial AUC as the area under each graph in the FPR range [0,0.2]. Note that if $fpr > 0.2$, then the partial AUC is 0. We report the average partial AUC over all 30 datasets for each algorithm.

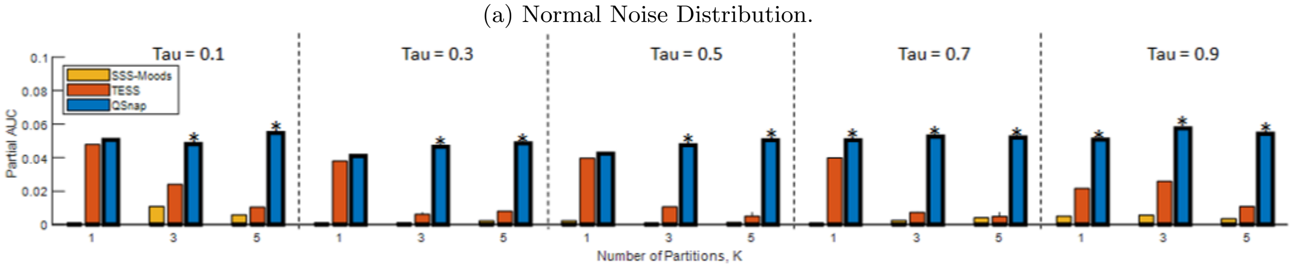
In our synthetic experiments, the highest possible partial AUC score is 0.2, if TPR = 1. This value is extremely unlikely, since our true positive points are not perfectly grouped together. Any region that overlaps all true positive points will almost certainly overlap negative points as well.

1.4 Additional Simulation Experiments

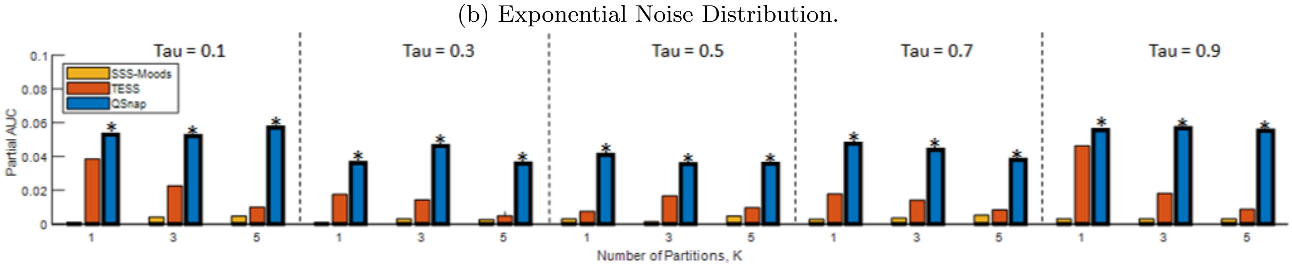
Figure 1 shows the full suite of experiments we performed with our data simulator, with $\tau = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ and $K = 1, 3, 5$. These additional experiments show the same trends as those reported in the main paper, with TESS performing poorly when $K > 1$, and SSS-Moods unable to do well in any setup.



(a) Normal Noise Distribution.



(b) Exponential Noise Distribution.



(c) Uniform Noise Distribution.

Figure 1: Partial AUC of TESS, SSS-Moods, and Qsnap on simulated data. The best performing algorithms are bolded, * indicates the best algorithm is statistically significant (Wilcoxon signed-rank test, $\alpha = 0.05$).