The Quantile Snapshot Scan: Comparing Quantiles of Spatial Data from Two Snapshots in Time

Travis Moore **Weng-Keen Wong**
School of EECS
Oregon State University

Abstract

We introduce the Quantile Snapshot Scan (Qsnap), a spatial scan algorithm which identifies spatial regions that differ the most between two snapshots in time. Qsnap is designed for spatial data with a numeric response and a vector of associated covariates for each spatial data point. Qsnap focuses on differences involving a specific quantile of the data distribution. A naive implementation of Qsnap is too computationally expensive for large datasets but our novel incremental update provides an order of magnitude speedup. We demonstrate Qsnap's effectiveness over an extensive set of experiments on simulated data. In addition, we apply Qsnap to two real-world problems: discovering bird migration paths and identifying regions with dramatic changes in drought conditions.

1 Introduction

Suppose an analyst is comparing counts of a specific species between the current year and the previous year. Each data point represents a geographic location with a response (e.g. the number of individuals observed) and an associated vector of covariates (e.g. features related to the observation process such as the time of day, time spent observing, etc.). The spatial aspect of the data is important, as is how the distribution of the response varies between locations. A common task when comparing spatial data from two different time periods is to look for the spatial region that is the most different between the two snapshots in time. In addition, the analyst may be interested in regions that differ according to a specific quantile of the response value. For example, the analyst may be interested in areas of high density for the species, such as those in the 75th percentile, and thus focus on how these regions have changed between the two snapshots.

This type of spatial analysis is applicable to many other spatial datasets, such as data from crop yields, property tax assessments and unemployment surveys. To solve problems of this nature, we introduce a novel algorithm called the Quantile Snapshot Scan (Qsnap for short) which is based on the Spatial Scan Statistic (SSS) (Kulldorff, 1997) framework. The original SSS was intended for purely spatial data (i.e. from a single snapshot in time). The SSS algorithm searches for the highest scoring region according to a hypothesis test and then computes a p-value characterizing the unusualness of that region. Different from the SSS, Qsnap finds the most different spatial region between two time periods, where the difference is measured relative to a specific quantile of the response variable. More precisely, Qsnap looks for differences relative to the two models predicting the conditional quantile function for the two snapshots. We show that Qsnap is more robust at detecting quantile differences for a variety of distributions than competing approaches, and we develop an efficient incremental update that speeds up a naive implementation of the algorithm by an order of magnitude. We also apply Qsnap to the tasks of identifying bird migration routes and detecting changes in drought conditions.

2 Background and Related Work

2.1 Quantile Regression

The τth quantile[1] (where $0 \leq \tau \leq 1$) of a continuous random variable Y is defined as the value q_τ such that

Proceedings of the 23rdInternational Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

[1]We will also refer to the pth percentile, which is equivalent to the $p/100$th quantile

$$q_\tau = \inf_y \{F(y) \geq \tau\} = F^{-1}(\tau) \quad (1)$$

where $F(Y) = P(Y \leq y)$ is the cumulative distribution function of Y. Intuitively, q_τ is the value where the proportion τ of the Y values are less than q_τ and $(1-\tau)$ of the Y values are greater than q_τ.

Just as least squares regression estimates the conditional mean function $E[\boldsymbol{Y}|\boldsymbol{X}] = \boldsymbol{X}\boldsymbol{\beta}$, quantile regression (Koenker and Bassett, 1978) fits the τth conditional quantile function $Q_{\boldsymbol{Y}}(\tau|\boldsymbol{X}) = \boldsymbol{X}\boldsymbol{\beta}(\tau)$. The parameters of the regression line are defined by solving:

$$\hat{\boldsymbol{\beta}}(\tau) = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} p_\tau(y_i - \boldsymbol{x_i}\boldsymbol{\beta}) \quad (2)$$

where $p_\tau(r) = r(\tau - I(r < 0))$ and $(y_i, \boldsymbol{x_i})$ is the ith data instance. This optimization finds a value of $\hat{\boldsymbol{\beta}}(\tau)$ such that the proportion τ of the total residuals are negative, and the proportion $(1-\tau)$ of the total residuals are positive. For notational convenience, we will drop the τ parameter on $\boldsymbol{\beta}$.

Quantile regression provides more flexibility and robustness than mean-based regression. τ can be set to focus the analysis on the quantile of interest, reducing the influence of other parts of the data distribution. For example, τ can be set close to 1 or 0 to model the extreme values of the distribution, or to 0.5 to fit the median. Distribution quantiles are also inherently more robust than means, which can be greatly influenced by extreme values (Rousseeuw and Leroy, 1987).

2.2 Rank Test for Quantile Regression

A key part of our algorithm requires comparing two quantile regression models using a hypothesis test. If we decompose the regression model at the τth quantile as $Q_{\boldsymbol{Y}}(\tau|\boldsymbol{X}, \tilde{\boldsymbol{X}}) = \boldsymbol{X}\boldsymbol{\beta}_1 + \tilde{\boldsymbol{X}}\boldsymbol{\beta}_2$, where $\boldsymbol{X}$ and $\tilde{\boldsymbol{X}}$ are each a set of covariates, then we can look at the hypothesis test $H_0 : \boldsymbol{\beta}_2 = \boldsymbol{0}$ vs $H_a : \boldsymbol{\beta}_2 \neq \boldsymbol{0}$. For the region detection task, $\tilde{\boldsymbol{X}}$ is an indicator of whether the data is from the region of interest or not. The test then equates to determining if data from the region follows the same model as the rest of the dataset.

One way to perform this hypothesis test is using the rank test for quantile regression (Gutenbrunner et al., 1993), which takes the form $T = \boldsymbol{S}'\boldsymbol{M}^{-1}\boldsymbol{S}$, where $\boldsymbol{S}$ is the score vector and $\boldsymbol{M}$ is the Fischer information matrix. For the hypothesis test defined above, the information matrix takes the form $\boldsymbol{M} = n^{-1}(\tilde{\boldsymbol{X}} - \boldsymbol{H}\tilde{\boldsymbol{X}})'(\tilde{\boldsymbol{X}} - \boldsymbol{H}\tilde{\boldsymbol{X}})$ with $\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})\boldsymbol{X}'$.

The score vector is the gradient of the log-likelihood. When the likelihood is unknown, the score can be approximated by a rank-score process. This assigns a best fit ranking to the datapoints as an empirical substitute for their probability. In our application, this ranking can be assigned with respect to the quantile of interest. For quantile regression, Machado and Silva (2002) approximate $\boldsymbol{S}$ as $\boldsymbol{S} = n^{-1/2}(\tilde{\boldsymbol{X}} - \boldsymbol{H}\tilde{\boldsymbol{X}})'\hat{\boldsymbol{b}}$.

The vector $\hat{\boldsymbol{b}}$ captures the rank score resulting from fitting a quantile regression at the τth quantile under H_0. Using the value of β_1 learned from H_0, each point is assigned a ranking value between τ and $\tau-1$ depending on whether the point falls above, below, or on the regression line. More specifically, $\hat{\boldsymbol{b}}_i = \hat{\boldsymbol{a}}_i(\tau) - (1-\tau)$ where $\hat{\boldsymbol{a}}_i = 1$ if $\boldsymbol{x_i}\boldsymbol{\beta_1} > 0$, $\hat{\boldsymbol{a}}_i = 0$ if $\boldsymbol{x_i}\boldsymbol{\beta_1} < 0$, and $0 \leq \hat{\boldsymbol{a}}_i \leq 1$ if $\boldsymbol{x_i}\boldsymbol{\beta_1} = 0$, subject to the constraint $\boldsymbol{X}'\hat{\boldsymbol{a}} = (1-\tau)\boldsymbol{X}'\boldsymbol{1}$. The values $\hat{\boldsymbol{a}}$ are the dual solution of the quantile regression optimization.

Our final test statistic is $T = \boldsymbol{S}'\boldsymbol{M}^{-1}\boldsymbol{S}/\Psi^2$, where $\Psi^2 = \tau(1-\tau)$ is included to normalize the score function used to compute $\hat{\boldsymbol{b}}$. T follows a Chi-squared distribution under the null hypothesis with $p = |\boldsymbol{\beta}_2|$ degrees of freedom. The rank test has the same asymptotic power as the analogous Wald and likelihood ratio tests, but does not require estimating the parameters under alternative hypothesis H_a, which greatly reduces the computation load for our algorithm's inner loop.

2.3 Spatial Scan Statistic

The SSS (Kulldorff, 1997) detects regions in which the frequency of occurrence of an event of interest is significantly different from expected. Events are modelled as a Poisson distribution where q is the probability of an event occurring. A scanning window search is performed over the dataset, with each window representing a subset of the data. For each data subset $\boldsymbol{C}$, a likelihood ratio test is used to test the null hypothesis that $q(\boldsymbol{C})$ (the probability of an event in local region $\boldsymbol{C}$), is equal to the global probability q, versus the alternative $q(\boldsymbol{C}) > q$. The SSS returns the most unusual window found, as determined by the p-value of the hypothesis test. Due to multiple hypothesis testing, the SSS uses a randomization test to produce an adjusted p-value for the most unusual region.

In theory, the search over data subsets should be exhaustive, but in practice it is often performed on a smaller number of subsets of a specific form, such as circular regions of expanding radius or rectangles of increasing size. Most variants of the SSS can be decomposed into a search method over candidate regions, and a test quantifying the unusualness of each region.

We briefly mention the most related SSS variants. The Fast Subset Scan algorithm (Neill, 2012) finds the most unusual subset of the data quickly if there exists an ef-

ficiently computed priority function to rank each data point. While very fast, the subset-scan algorithm does not guarantee that the subset is spatially contiguous, and the corresponding priority function for quantile regression does not meet its assumptions. The space-time scan statistic (Kulldorff et al., 1998) finds unusual "cylinders" in space-time relative to the entire dataset. This approach fundamentally differs from our setup, where the same spatial area is compared at two different times to discover local changes.

Unlike most SSS variants, the Treatment Effect Spatial Scan (TESS) (McFowland et al., 2018) tests the τth quantile to find unusual regions rather than the distribution means. The TESS algorithm divides the data into a control and treatment group. It fits a model to the control group, and uses it to produce a p-value for every point in the treatment group. A subset-scan algorithm is used to find a discrete set of features in the treatment group where the p-values are most different from the expected value. For a given subset $\boldsymbol{C}$ of the treatment group, it computes the likelihood ratio test statistic $T = |\boldsymbol{C}|\frac{(\tilde{\tau}-\tau)^2}{\tau(1-\tau)}$, where $\tilde{\tau}$ is the proportion of p-values in $\boldsymbol{C}$ less than τ. The search is performed on a range of quantiles $[\tau-\alpha, \tau+\alpha]$, with the best overall region reported. The treatment/control setup of TESS can be adapted to our problem of looking for differences between two snapshots in time, thereby making TESS the closest related work to Qsnap.

Finally, the Quantile Spatial Scan Statistic (QSSS) (Moore and Wong, 2018) operates on spatial data from a single time slice and finds unusual spatial regions relative to the overall space. QSSS uses an efficient incremental rank test on quantile regression to find unusual regions at the τth quantile. A key factor in its computational speed is that regardless of the candidate region under consideration, under the null hypothesis, the total comparison space $\boldsymbol{X}$ does not change. This assumption is violated in the space-time setting of Qsnap, where points outside of candidate region $\boldsymbol{C}$ are not considered in the test, causing the comparison space to change as $\boldsymbol{C}$ changes. Without its speedup, QSSS is intractable to run on even moderate sized datasets. Consequently, the incremental update for Qsnap addresses a fundamentally different optimization problem than the incremental update for QSSS.

2.4 Other Related Work

The computer vision and remote sensing communities address a seemingly related problem of change detection for images and landsat data taken at different times (see Radke et al. (2005), Zhu (2017) for surveys). This line of research is different from our work as the techniques are specifically intended for images and regularly gridded landsat values, rather than randomly located spatial data. In addition, these methods do not compare quantiles of the data distributions.

Many statistical techniques have been developed to produce a smooth surface representing a specific quantile of a spatial distribution (e.g. Hallin et al. (2009); Reich et al. (2011); Lum and Gelfand (2012); Macmillan (2013)). One could apply these methods to fit models to the data from the two snapshot time periods and then take a difference between these surfaces. This difference would produce an informative visualization but identifying a specific region that differs the most between the two time periods would need to be done manually; this manual inspection is time consuming, especially for a big region, or it can be done somewhat clumsily with an additional post-processing algorithm, which would resemble the search step in our Qsnap algorithm. The Qsnap algorithm not only performs this analysis in one holistic framework, but it is also more computationally efficient and it enables an automated monitoring system that avoids the need for human inspection of the difference surface.

3 Methodology

Suppose we are given two spatial datasets obtained at two different times (i.e. snapshots). We denote the two datasets as $\boldsymbol{D}^{(1)} = \{\boldsymbol{Y}^{(1)}, \boldsymbol{X}^{(1)}, \boldsymbol{L}^{(1)}\}$ and $\boldsymbol{D}^{(2)} = \{\boldsymbol{Y}^{(2)}, \boldsymbol{X}^{(2)}, \boldsymbol{L}^{(2)}\}$. For the dataset at snapshot s, we denote the ith data point as $\boldsymbol{D}_i^{(s)} = (y_i^{(s)}, \boldsymbol{x}_i^{(s)}, \boldsymbol{l}_i^{(s)})$ where $y_i^{(s)}$ is the continuous response, $\boldsymbol{x}_i^{(s)} = (x_{i,1}^{(s)}, \ldots, x_{i,p}^{(s)})$ are the p covariates associated with the ith data point and $\boldsymbol{l}_i^{(s)} = (l_{i,1}^{(s)}, \ldots, l_{i,d}^{(s)})$ are the d dimensional coordinates specifying the spatial location of the data point. If $d = 2$, the location tuple $(l_{i,1}^{(s)}, l_{i,2}^{(s)})$ can represent latitude and longitude. Note that the set of locations $\boldsymbol{L}^{(1)}$ and $\boldsymbol{L}^{(2)}$ are not required to be from the exact same locations for the two snapshots, but they should be from the same general region. Our goal is to find a region $\boldsymbol{C}$ which is the most different between the two snapshots (with respect to the τth quantile of the response variable) and to compute a score that characterizes this region's unusualness.

The Qsnap algorithm searches over candidate regions $\boldsymbol{C}$. As is commonly done in SSS variants, the search involves looking at regions of expanding size (e.g. circle with increasing radius), centered at evenly spaced gridpoints covering the space $\boldsymbol{L} = \boldsymbol{L}^{(1)} \cup \boldsymbol{L}^{(2)}$. For each candidate region, Qsnap performs a hypothesis test to determine when the τth quantile of $\boldsymbol{D}^{(1)}$ is different from the τth quantile of $\boldsymbol{D}^{(2)}$ in region $\boldsymbol{C}$. For this paper, we will focus on optimizing the speed and power of the hypothesis test.

3.1 Snapshot Hypothesis Test

Given a region $\boldsymbol{C}$, we denote the response variables and associated covariates of the datapoints from snapshot s in region $\boldsymbol{C}$ as $\boldsymbol{Y}_{\boldsymbol{C}}^{(s)}$ and $\boldsymbol{X}_{\boldsymbol{C}}^{(s)}$ respectively. We want to compare $Q_{\boldsymbol{Y}_{\boldsymbol{C}}^{(1)}}(\tau|\boldsymbol{X}_{\boldsymbol{C}}^{(1)})$ and $Q_{\boldsymbol{Y}_{\boldsymbol{C}}^{(2)}}(\tau|\boldsymbol{X}_{\boldsymbol{C}}^{(2)})$, the τth quantile regression models of $\boldsymbol{C}$ at snapshots 1 and 2. We define $\boldsymbol{X} = [\boldsymbol{X}_{\boldsymbol{C}}^{(1)}; \boldsymbol{X}_{\boldsymbol{C}}^{(2)}]$ and $\boldsymbol{Y} = [\boldsymbol{Y}_{\boldsymbol{C}}^{(1)}; \boldsymbol{Y}_{\boldsymbol{C}}^{(2)}]$, where the symbol ";" indicates matrix concatenation (as in Matlab). We construct the matrix $\tilde{\boldsymbol{X}}$ such that $\tilde{\boldsymbol{X}}_i = \boldsymbol{X}_i$ if point i is from snapshot 2, and $\boldsymbol{0}$ (i.e. the zero vector) otherwise. We can then set up the quantile regression model $Q_{\boldsymbol{Y}}(\tau|\boldsymbol{X}, \tilde{\boldsymbol{X}}) = \boldsymbol{X}\boldsymbol{\beta_1} + \tilde{\boldsymbol{X}}\boldsymbol{\beta_2}$. This represents a nested model where $\boldsymbol{\beta_1}$ are the parameters for snapshot 1 and $(\boldsymbol{\beta_1} + \boldsymbol{\beta_2})$ are the parameters for snapshot 2. Testing the null hypothesis $\boldsymbol{\beta_2} = \boldsymbol{0}$ vs the alternative $\boldsymbol{\beta_2} \neq \boldsymbol{0}$ will determine if the region $\boldsymbol{C}$ in snapshot 2 is significantly different from the region in snapshot 1 at the τth quantile.

We choose to use the rank test for quantile regression, because it only requires fitting a quantile regression model for $\boldsymbol{\beta_1}$ under the null hypothesis, and has the same asymptotic power as a likelihood ratio test. The test statistic T can be formulated as $T = \hat{\boldsymbol{b}}'\boldsymbol{Z}(\boldsymbol{Z}'\boldsymbol{Z})^{-1}\boldsymbol{Z}'\hat{\boldsymbol{b}}/\Psi^2$ where $\boldsymbol{Z} = \tilde{\boldsymbol{X}} - \boldsymbol{H}\tilde{\boldsymbol{X}}$.

Alternatively, we can write this as $T = \hat{\boldsymbol{b}}'\boldsymbol{Q_Z}\boldsymbol{Q}_{\boldsymbol{Z}}'\hat{\boldsymbol{b}}/\Psi^2$ if $\boldsymbol{Q_Z}$ is an orthonormal basis for $\boldsymbol{Z}$. In either form, re-calculating T from scratch each time $\boldsymbol{C}$ changes can be very time consuming. Moore and Wong (2018) proposed a speedup for a quantile spatial scan algorithm leveraging this formulation, under the assumptions that $\boldsymbol{X}$, $\boldsymbol{H}$, $\hat{\boldsymbol{b}}$, and $\boldsymbol{\beta_1}$ are all constant between iterations. These assumptions are violated in the snapshot scan setting. Specifically, when $\boldsymbol{C}$ grows by one point, both $\tilde{\boldsymbol{X}}$ and $\boldsymbol{X}$ grow by an additional row, requiring $\boldsymbol{H}$, $\hat{\boldsymbol{b}}$, and $\boldsymbol{\beta_1}$ to be recalculated. In the following section we will outline a novel speedup for the snapshot scan setting.

3.2 An Efficient Incremental Update

We now derive a novel incremental update for Qsnap which drastically reduces the time needed to calculate T when $\boldsymbol{C}$ grows. In each iteration of Qsnap, the size of the region $\boldsymbol{C}$ grows by one point, which increases the size of $\boldsymbol{X}$ and $\tilde{\boldsymbol{X}}$ by one row and requires a recalculation of the test statistic T for the new region. Changes to $\boldsymbol{X}$ and $\tilde{\boldsymbol{X}}$ means $\boldsymbol{Q_Z}$, $\boldsymbol{Z}$, $\boldsymbol{H}$, and $\hat{\boldsymbol{b}}$ must be updated to re-calculate T. While T only depends on $\boldsymbol{Q_Z}$ and $\hat{\boldsymbol{b}}$, $\boldsymbol{Q_Z}$ will change in relation to $\boldsymbol{Z}$, since it is a basis for $\boldsymbol{Z}$, and $\boldsymbol{Z}$ depends on $\boldsymbol{H}$. In the following sections, we describe efficient incremental updates for these values. For many values, we save time by updating the QR decomposition of the data matrix, instead of the matrix itself. We use the superscript t to indicate the current iteration, and $t+1$ the iteration after adding a new data point. Proofs of the theorems used can be found in the supplemental material.

Updating $\boldsymbol{H}$: We start by addressing how to quickly update $\boldsymbol{H}^t$ as $\boldsymbol{X}^t$ grows in size. First, let $\boldsymbol{X}^t = \boldsymbol{Q}_{\boldsymbol{X}}^t\boldsymbol{R}_{\boldsymbol{X}}^t$ be the QR factorization of $\boldsymbol{X}^t$. Note that $\boldsymbol{H}^t = \boldsymbol{X}^t(\boldsymbol{X}^{t\prime}\boldsymbol{X}^t)^{-1}\boldsymbol{X}^{t\prime}$ is a projection matrix, and can be re-written as $\boldsymbol{H}^t = \boldsymbol{Q}_{\boldsymbol{X}}^t\boldsymbol{Q}_{\boldsymbol{X}}^{t\prime}$ since $\boldsymbol{Q}_{\boldsymbol{X}}^t$ is an orthonormal basis for $\boldsymbol{X}^t$. An initial $\boldsymbol{Q_X}$ comes from the Q matrix of the QR factorization of $\boldsymbol{X}^t$ at $t = 1$.

Theorem 1 (supplemental) shows that when a new row is appended to $\boldsymbol{X}^t$, then

$$\boldsymbol{H}^{t+1} = \boldsymbol{Q}_{\boldsymbol{X}}^{t+1}\begin{bmatrix}1 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{H}^t\end{bmatrix} - \boldsymbol{v}\boldsymbol{v}' \tag{3}$$

Where $\boldsymbol{v}$ is the last column of $\boldsymbol{Q}_{\boldsymbol{X}}^{t+1}$. $\boldsymbol{Q}_{\boldsymbol{X}}^{t+1}$ can be efficiently found using the algorithm in Section 12.5.3 of Golub and Van Loan (2012), which uses $O(p)$ Givens rotation to update the existing QR factorization.

Updating $\boldsymbol{Z}$: If $\boldsymbol{Z}^t$ and $\boldsymbol{Q}_{\boldsymbol{X}}^{t+1}$ are known, then using Theorem 2 (supplemental) we have

$$\boldsymbol{Z}^{t+1} = \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{Z}^t\end{bmatrix} + \boldsymbol{v}\boldsymbol{g}' \text{ where } \boldsymbol{g} = [\tilde{\boldsymbol{x}}^{t+1}, \tilde{\boldsymbol{X}}^t]\boldsymbol{v} \tag{4}$$

The incremental update to $\boldsymbol{Z}^t$ has two simple steps: append a zero row, and add the rank one matrix $\boldsymbol{v}\boldsymbol{g}'$. Explicitly calculating $\boldsymbol{H}^{t+1}$ to find $\boldsymbol{Z}^{t+1}$ is not needed as computing $\boldsymbol{Q}_{\boldsymbol{X}}^{t+1}$ and then reading off its last column to produce $\boldsymbol{v}$ is sufficient.

Updating $\boldsymbol{Q_Z}$: Theorem 3 (supplemental) shows that, given an initial QR decomposition $\boldsymbol{Z}^t = \boldsymbol{Q}_{\boldsymbol{Z}}^t\boldsymbol{R}_{\boldsymbol{Z}}^t$, the factorization at iteration $t+1$ is

$$\boldsymbol{Q}_{\boldsymbol{Z}}^{t+1} = \boldsymbol{Q}^t\boldsymbol{G}_{\boldsymbol{R}}'\boldsymbol{G}_{\boldsymbol{B}}' \tag{5}$$

$$\boldsymbol{R}_{\boldsymbol{Z}}^{t+1} = \boldsymbol{G_B}\boldsymbol{G_R}\Big(\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{R}_{\boldsymbol{Z}}^t\end{bmatrix} + \boldsymbol{c}\boldsymbol{g}'\Big) \tag{6}$$

where $\boldsymbol{G_R}$ and $\boldsymbol{G_B}$ are both computed by multiplying $O(p)$ Givens rotation matrices. The proof uses the algorithm in section 12.5.1 of Golub and Van Loan (2012) for rank one updates of QR decompositions.

Updating $\hat{\boldsymbol{b}}$: $\hat{\boldsymbol{b}}$ is based on the dual solution to the quantile regression, and can be directly computed if the primal solution $\boldsymbol{\beta}_1$ is known. $\boldsymbol{\beta}_1$ can be calculated more efficiently by warmstarting the optimization algorithm (i.e starting the optimization at the previous

solution point). Since the changes to $\boldsymbol{Y}$ and $\boldsymbol{X}$ are small, we can expect the previous solution to be relatively close to the new solution. We use the simplex method as the optimization algorithm, because pivot operations are fast, and the algorithm is very efficient when started near the optimal solution.

Algorithm 1 shows the entire update process when point j is added to $\boldsymbol{C}$. The functions rowUpdate() and rankOneUpdate() correspond to the QR update algorithms from Golub and Van Loan (2012). simplexSolve() uses the simplex algorithm to compute a quantile regression solution, warmstarted at a given solution value. dualSolution() returns the dual solution to the quantile regression given the primal solution.

Both rowUpdate() and rankOneUpdate() can be run in $O(np)$ time, as their main bottleneck is the multiplication of $O(p)$ Givens rotations; the rotation matrices are sparse, so each can be done in $O(n)$ time. The quantile regression dual solution can be calculated in $O(np)$ time when the primal solution is known. For the simplex solver, each pivot operation takes $O(np)$ time. Convergence is guaranteed by the convex solution space, but in general we cannot say how many pivots will be required. Provided the data is reasonably well behaved, warmstarting should significantly reduce the number of pivots required, especially when adding a single new data point. In practice, we have observed that the warmstarted simplex algorithm often converges in a sublinear number of pivots.

3.3 Multiple Hypothesis Test Correction

With multiple hypothesis tests being performed, we cannot use the rank test p-value to determine the significance of the most extreme region. Instead, we apply a Gumbel correction (Abrams et al., 2010), which uses the most significant test values from randomized data permutations to parameterize an extreme value distribution and identify significant values. This approach requires less data permutations than the traditional randomization test. If multiple significant regions need to be returned, methods like the false discovery rate (Benjamini and Hochberg, 1995) or Bonferroni correction (Bonferroni, 1936), can be used.

4 Results

Evaluating Qsnap on real-world data is challenging due to the lack of datasets with ground-truth identification of which region(s) change (or did not change) from one time period to another. Consequently, we create simulated data where the ground truth is known and we perform an extensive set of experiments under different simulator settings. We also compare Qsnap against other algorithm on two real-world problems and we corroborate the results against findings by independent sources.

Algorithm 1 Incremental Rank Test

Inputs: $\boldsymbol{Y}, \boldsymbol{X}, \tilde{\boldsymbol{X}}, \boldsymbol{x}^{t+1}, \tilde{\boldsymbol{x}}^{t+1}, \boldsymbol{Q_X}, \boldsymbol{R_X}, \boldsymbol{Q_Z}, \boldsymbol{R_Z}, \beta_1, \tau$
$[\boldsymbol{Q_X}, \boldsymbol{R_X}] = \text{rowUpdate}(\boldsymbol{Q_X}, \boldsymbol{R_X}, \boldsymbol{x}^{t+1})$
$\boldsymbol{v} = \boldsymbol{Q_X}.lastColumn$
$\boldsymbol{Q_X} = \text{removeLastColum}(\boldsymbol{Q_X})$
$\tilde{\boldsymbol{X}} = \begin{bmatrix} \tilde{\boldsymbol{x}}^{t+1} \\ \tilde{\boldsymbol{X}} \end{bmatrix}$
$\boldsymbol{g} = \tilde{\boldsymbol{X}}' v$
$[\boldsymbol{Q_Z}, \boldsymbol{R_Z}] = \text{rowUpdate}(\boldsymbol{Q_Z}, \boldsymbol{R_Z}, \boldsymbol{0})$
$[\boldsymbol{Q_Z}, \boldsymbol{R_Z}] = \text{rankOneUpdate}(\boldsymbol{Q_Z}, \boldsymbol{R_Z}, \boldsymbol{v}, \boldsymbol{g}')$
$\beta_1 = \text{simplexSolve}(\boldsymbol{X}, \boldsymbol{Y}, \tau, \beta_1)$
$\boldsymbol{a} = \text{dualSolution}(\boldsymbol{X}, \boldsymbol{Y}, \tau, \beta_1)$
$\hat{\boldsymbol{b}} = \boldsymbol{a} - (1 - \tau)$
$T = \hat{\boldsymbol{b}}' \boldsymbol{Q_Z} \boldsymbol{Q}'_{\boldsymbol{Z}} \hat{\boldsymbol{b}} / (\tau(1-\tau))$
Return$(T, \tilde{\boldsymbol{X}}, \boldsymbol{Q_X}, \boldsymbol{R_X}, \boldsymbol{Q_Z}, \boldsymbol{R_Z}, \beta_1)$

4.1 Runtime Analysis

We compare our rank test speedup against two baselines. The first baseline is a naive implementation that calculates the test statistic T from scratch every iteration; the second is a naive implementation that uses warmstarting to re-learn $\boldsymbol{\beta}_1$ quickly each iteration.

Table 1 shows the average update time of each algorithm as the size of the dataset increases, with the number of features constant at $p = 3$. Our incremental algorithm is by far the fastest for larger n, demonstrating a linear increase in runtime while the others increase quadratically. The difference between the naive and warmstarted algorithms is relatively small, showing that the primary computational bottleneck is in calculating $\boldsymbol{Z}(\boldsymbol{Z}'\boldsymbol{Z})^{-1}\boldsymbol{Z}'$.

4.2 Simulation Experiments

We evaluate the accuracy of Qsnap on simulated data where the changed regions are known. We compare Qsnap against two other quantile spatial scan algorithms, with the first being a variant of the Treatment-Effect Spatial Scan (TESS) (McFowland et al., 2018). We adapt TESS to the snapshot scan setting by using the first and second snapshots as the "control" and "treatment" groups, respectively. In the original paper, p-values for each data point are calculated non-parametrically as the ratio of points with a higher response value. For our multi-variate data, we compute a quantile regression on the control set at the desired quantile, compute the residuals for this regression on the treatment set, then find the non-parametric p-

n	1000	2000	4000	6000	8000
Naive	1.8	6.1	20.9	48.6	81.4
Warmstart	1.1	4.4	17.6	43.0	74.9
Incremental	**0.3**	**0.4**	**0.8**	**1.1**	**1.4**

Table 1: Average test statistic update time in ms, averaged over 1000 updates for randomly generated data.

values of those residuals. We found this approach produces far better results than assuming a parametric distribution form. Like Qsnap, TESS can find subsets of the second snapshot that most differ from their expected distribution with respect to a specific quantile. We replace the discrete subset search from the original TESS paper with the same search used by Qsnap, for a more direct comparison of the hypothesis tests.

To our knowledge, TESS is the only existing algorithm that can be directly applied to our task. We create a second baseline using the SSS framework. First, we modify the SSS to search for the most significant region between two snapshots in time. Second, we replace the likelihood ratio test of the SSS with Mood's hypothesis test (Mood, 1950) so that we can compare the τth quantiles of the regions under consideration. For a given region $\boldsymbol{C}$, this test fits the quantile regression $Q_{\boldsymbol{Y}_{\boldsymbol{C}}^{(1)}}(\tau|\boldsymbol{X}_{\boldsymbol{C}}^{(1)})$, then performs a 2×2 chi-squared test on the number of points above and below the regression line in snapshot one and snapshot two. We refer to this algorithm as SSS-Moods.

4.2.1 Simulator

Our simulator generates data points as a tuple $\{y_i, \boldsymbol{x}_i, \boldsymbol{l}_i\}$. The location ($\boldsymbol{l}_i$) and parameter values ($\boldsymbol{x}_i$) are created uniformly at random between a set of maximum and minimum values. Each dataset is partitioned into K spatially contiguous sections. For each section k, the response, y_i, is calculated from a linear relationship $y_i = \boldsymbol{\beta}^k \boldsymbol{x}_i + \epsilon$, where $\boldsymbol{\beta}^k$ is a randomly generated parameter vector that is different for each section, and ϵ is a random noise term. In our experiments, we compare normal, exponential, and uniform distributions for the random noise term ϵ. Our simulator models data that is globally heterogeneous, but homogeneous for specific local spatial areas.

Two data snapshots of n points each are generated using the same partition boundaries and values of $\boldsymbol{\beta}^1, \ldots, \boldsymbol{\beta}^K$. In the second snapshot, a partition j is designated as the target area. In the target area, a subset of the response values are generated from a shifted distribution as $y_i = (\boldsymbol{\beta}^j + \boldsymbol{\delta})\boldsymbol{x}_i + \epsilon$, where $\boldsymbol{\delta}$ is a random vector with $||\boldsymbol{\delta}|| = p$. When generating data, we can identify which quantile q_i of the error distribution each data point falls into. Any point in the target area with a quantile value q_i such that $|\tau - q_i| \leq 0.1$ is generated from this shifted distribution with parameters $\boldsymbol{\beta}^j + \boldsymbol{\delta}$, while the rest are generated with $\boldsymbol{\beta}^j$. This effectively creates a change in 20% of the distribution in the target area, centered around the τth quantile; detecting this change in the simulated datasets is in general a very challenging problem. A successful algorithm will find the area j by performing tests at the τth quantile.

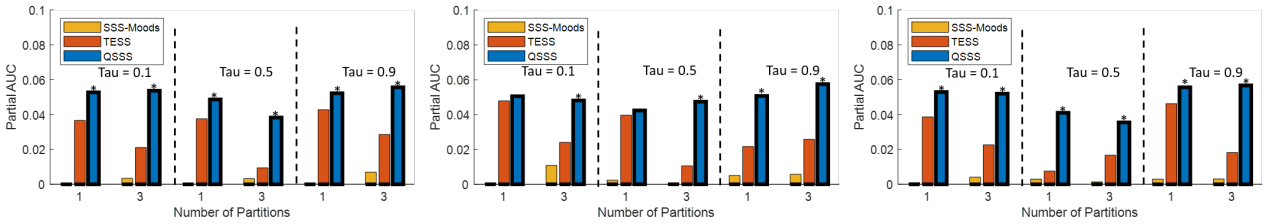
4.2.2 Simulation Results

We performed a suite of experiments to compare the performances of Qsnap, TESS, and SSS-Moods. In each experiment, each algorithm is tasked with finding the target area in snapshot two where 20% of the points are generated from the shifted distribution. Each algorithm performs its search on the τth quantile, the center of the distribution shift. Each algorithm uses the same spatial search routine, allowing a direct comparison of their hypothesis tests. We evaluate each algorithm's performance by the most significant area reported by each algorithm on each dataset.

We tested each algorithm on simulated data with normal, exponential, and uniform noise distributions, with distribution changes centered at quantiles $\tau = 0.1, 0.5, 0.9$ in the target area. We also varied the number of partitions K between 1 and 3. For $K = 1$, the snapshots have the same base distribution at all locations, and the target region is a random circle in $\boldsymbol{L}$.

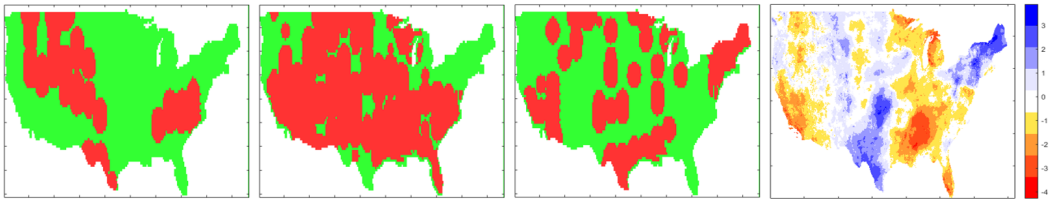
We evaluate the algorithms' true positive rate (TPR) versus false positive rate (FPR) curve, calculated on a per-point basis. Only points generated from the shifted distribution are counted as true positives. Typically, a good summary of this curve can be captured with the area under the curve (AUC). In a real-world setting, we would never realistically operate the algorithm under a high false positive rate and the AUC for high FPR values is not meaningful. As a result, we use the partial AUC to measure performance; specifically, we report the AUC from $FPR = [0, 0.2]$ to emphasize lower FPR values. We compute the partial AUC for each algorithm on each dataset, and report the average value across 30 randomly generated datasets. See the supplemental for calculation details.

Figure 1 shows the average partial AUCs for each algorithm at different values of τ, K, and different forms of the noise distribution. These experiments were run for $n = 5000$ and $p = 5$. The size of the target area in snapshot two was set at 1000 points, meaning approximately 200 points generated from the shifted distribution. We do not show experiments with changing values of n, p, or target area size, as these parameters affected each algorithm in similar, intuitive ways. Experiments with $\tau = 0.3, 0.7$ and $K = 5$ illustrated the



(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, $\alpha = 0.05$).



(a) TESS (b) SSS-Moods (c) Qsnap (d) Change in SPEI

Figure 2: (a)-(c) Significant areas found on climate data. (d) Change in SPEI between 2001 and 2007.

same trends, and can be found in the supplemental.

SSS-Moods performs very poorly in these experiments, barely detecting any changes. The low-power Mood's quantile test is ill-suited to the difficulty of this problem. TESS does well for $K = 1$, but poorly on larger K. The speed of TESS is dependent on fitting a global model instead of many local models, making it ill-suited for data with local variation.

Qsnap performs significantly better than the other algorithms in 16/18 of the experiments. Like SSS-Moods, it fits a model to each local region, allowing it to account for spatially varying distributions. It also uses a more powerful test statistic, giving it significantly greater accuracy than the other algorithms.

The runtimes of the algorithms, averaged over 10 datasets ($n = 5000$, $p = 5$, and $\tau = 0.7$) are: 87 seconds (Qsnap), 25 seconds (SSS-Moods) and 0.2 seconds (TESS). SSS-Moods and TESS perform simpler hypothesis tests and are faster than Qsnap. However, the simpler tests cause them to be unable to detect many changed regions and they lack robustness. In the next section, we show that both TESS and SSS-Moods also perform poorly on real-world data.

4.3 Drought Detection

We use Qsnap, TESS, and SSS-Moods to detect changes in drought conditions in the continental US using climate data collected from `climateengine.org` (Huntington et al., 2017). Our model uses precipitation as the response, with humidity, evaporation, mean temperature, max temperature, and soil moisture (5cm level) as covariates. We use 2001, a relatively mild drought year for most of the country, and 2007, which had extensive droughts in California and the South, as our two time snapshots to compare.

Each algorithm is tasked with finding areas of significant change between the two years at the 10th percentile. Tuning the algorithms to a low percentile of precipitation makes them better suited to finding changes in drought conditions. Since our dataset contains many areas of significant drought change, each algorithm reports all significant regions instead of the most significant, using the Holm–Bonferroni correction with $\alpha = 0.01$ (Holm, 1979). Though the observations in this data are in a consistent grid structure, our algorithm does not require such conditions to be used.

To evaluate the regions returned by each algorithm we use the Standardised Precipitation-Evapotranspiration Index (SPEI). SPEI is a numerical measure of drought severity, calculated based on the difference of precipitation and potential evaporation. These SPEI values are independent from our climate data, and we use the difference in SPEI between 2007 and 2001 as a proxy for the (unknown) ground truth.

In Table 2 we report three evaluation metrics. First is $\Delta(C)$, the average change in SPEI for observations in each detected region C, using absolute difference of each observation. An algorithm that more accurately detects regions of high change will have a higher value. The second metric is the average absolute change in all regions not detected by the algorithm, $\overline{C}$, which

	Avg. $\Delta(C)$	$\Delta(\overline{C})$	Avg. $\Delta_\tau(C)$
TESS	0.76	0.95	0.66
SSS-Moods	0.87	0.95	0.86
Qsnap	**1.20***	**0.81**	**1.17***

Table 2: Evaluation of detected regions for climate data using change in SPEI on regions found, regions omitted, and the τth quantile change in regions found. * for statistical significance (paired t-test, $\alpha = 0.05$)

will be lower for better performers. We also report the average absolute change in the 10th percentile of each detected region, $\Delta_\tau(C)$, which should increase for better performers. Qsnap performs the best for all three algorithms in all of these categories.

Figure 2 shows the significant areas found by each algorithm, along with a heat map of the change in SPEI between 2007 and 2001. Red and orange areas in the heat map experienced increased drought conditions, while blue areas decreased. Qsnap finds many areas of significant change, while including less areas of no change. In comparison, TESS fails to find many of the most significant areas while including less significant ones, and SSS-Moods includes nearly the entire map. Note that we do not claim QSnap is the best tool for drought detection – only that it outperforms TESS and SSS-Moods.

4.4 Discovering Migration Paths in eBird

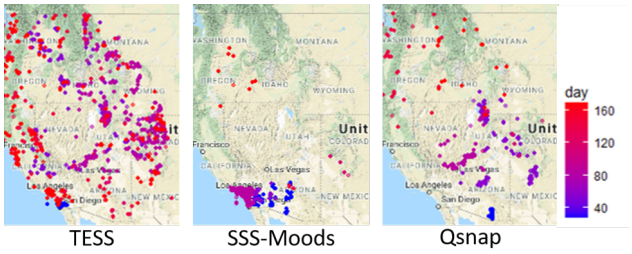


Figure 3: Most significant regions ($\tau = 0.9$) found on eBird data. Regions are color coded for time of year.

We applied Qsnap to eBird checklists (Sullivan et al., 2014), which record bird observations identified by species by citizen scientists. We aim to discover migration routes by identifying areas with an unusual influx of birds between time periods. By segmenting the migration period into snapshots, and running Qsnap on those snapshots, the anomalous regions found should discover the migration path. We identify unusual regions by finding changes in the 90th percentile of the number of birds reported on checklists in a region.

We restricted our dataset to the western flyway of the United States, during 2017. We only included checklists from that year, and in Bird Conservation Regions[2] (BCRs) 5, 9-10, 15-16, 32-35. Past work (La Sorte and Fink, 2017) has shown that migrating birds pass through this area during the first half of the year, traveling north from Central America and staying west of the Rocky Mountains. We divided the data from the first half of the year into 6 snapshots spanning 4 weeks each, and ran Qsnap, TESS, and SSS-Moods on pairs of consecutive snapshots. Note that, unlike other studies of migration patterns (e.g. La Sorte and Fink (2017)), we did not filter our dataset to only include migrating species or smooth the observations over space. Our dataset included every checklist for every species in the region and time frame, making this dataset very challenging due to the noise from non-migratory bird species and the imperfect observation process. We used time spent observing as the covariate, and number of individual birds seen as the response. For all algorithms, we used expanding rectangles instead of circles for the spatial scan search because rectangles can better detect longer regions that capture the South to North migration.

Figure 3 shows the areas[3] reported by each algorithm over the 6 snapshots of data from the first 168 days of 2017. The areas are represented by the individual checklists inside them, and color coded for time of year. The areas found by TESS were large and sporadic, with no clear pattern over time. The areas found by SSS-Moods seem to identify the start and end of the expected migration path, but fail to connect them well. Qsnap produced the best South to North gradient over time, identifying a clear progression that closely matches the migration route shown in La Sorte and Fink (2017). Being able to identify a shifting spatial trend in the complex and highly noisy eBird dataset without any data filtering illustrates the robustness and usefulness of Qsnap.

5 Conclusion

We presented Qsnap, which finds a region in spatial data that has undergone the most significant change between two data snapshots, with respect to the τth quantile. To reduce the computational cost of a search over multiple regions, we developed an efficient incremental update to the rank test that makes the algorithm scalable to large datasets. Compared to other similar algorithms, Qsnap is less dependent on assumptions about the data, and performs better on a variety of different distributions. Qsnap was also better suited to our two real-world data analysis tasks.

[2]See http://nabci-us.org/resources/bird-conservation-regions-map/ for the regions

[3]All maps generated using ggmaps in R (Kahle and Wickham, 2013)

Acknowledgements

This research was funded in part by NSF grant CCF-1521687. We thank Frank La Sorte and Daniel Fink for their expertise and feedback.

References

Abrams, A., Kleinman, K., and Kulldorff, M. (2010). Gumbel based p-value approximations for spatial scan statistics. *Int J Health Geogr*, 9(1):61.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B*, 57(1):289–300.

Bonferroni, C. E. (1936). Teoria statistica delle classi e calcolo delle probabilità. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*, 8:3–62.

Golub, G. and Van Loan, C. (2012). *Matrix computations*, volume 3. JHU Press, Baltimore, Maryland. Sections 12.5.1 and 12.5.3.

Gutenbrunner, C., Jureckova, J. K. R. S., Koenker, R., and Portnoy, S. (1993). Tests of linear hypotheses based on regression rank scores. *J Nonparametr Stat*, 2(4):307–331.

Hallin, M., Lu, Z., and Yu, K. (2009). Local linear spatial quantile regression. *Bernoulli*, 15(3):659–686.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70.

Huntington, J., Hegewisch, K., Daudert, B., Morton, C., Abatzoglou, J., McEvoy, D., and Erickson, T. (2017). Climate engine: Cloud computing of climate and remote sensing data for advanced natural resource monitoring and process understanding. *Bulletin of the American Meteorological Society*. `http://journals.ametsoc.org/doi/abs/10.1175/BAMS-D-15-00324.1`.

Kahle, D. and Wickham, H. (2013). ggmap: Spatial visualization with ggplot2. *The R Journ al*, 5(1):144–161. `http://journal.r-project.org/archive/2013-1/kahle-wickham.pdf`.

Koenker, R. and Bassett, G. (1978). Regression quantiles. *Econometrica*, 46(1):33–50.

Kulldorff, M. (1997). A spatial scan statistic. *Communications in Statistics: Theory and Methods*, 26(6):1481–1496.

Kulldorff, M., Athas, W. F., Feurer, E. J., et al. (1998). Evaluating cluster alarms: a space-time scan statistic and brain cancer in los alamos, new mexico. *American journal of public health*, 88:1377–1380.

La Sorte, F. A. and Fink, D. (2017). Migration distance, ecological barriers and en-route variation in the migratory behaviour of terrestrial bird populations. *Global Ecology and Biogeography*, 26:216–227.

Lum, K. and Gelfand, A. E. (2012). Spatial quantile multiple regression using the asymmetric laplace process. *Bayesian Analysis*, 7(2):235–258.

Machado, J. and Silva, J. M. C. S. (2002). Quantiles for counts. *J Am Stat Assoc*, 100(472):1226–1237.

Macmillan, D. P. (2013). *Quantile Regression for Spatial Data*. Springer-Verlag, Berlin.

McFowland, III, E., Somanchi, S., and Neill, D. B. (2018). Efficient Discovery of Heterogeneous Treatment Effects in Randomized Experiments via Anomalous Pattern Detection. arXiv preprint arXiv: 1803.09159.

Mood, A. (1950). *Introduction to the Theory of Statistics*. McGraw Hill Book Co., New York.

Moore, T. and Wong, W.-K. (2018). An efficient quantile spatial scan statistic for finding unusual regions in continuous spatial data with covariates. In *Uncertainty in Artificial Intelligence*, pages 756–765, Corvallis, OR. AUAI Press.

Neill, D. B. (2012). Fast subset scan for spatial pattern detection. *J R Stat Soc Series B Stat Methodol*, 74(2):337–360.

Radke, R. J., Andra, S., Al-Kofahi, O., and Roysam, B. (2005). Image change detection algorithms: a systematic survey. *IEEE Transactions on Image Processing*, 14:294–307.

Reich, B. J., Fuentes, M., and Dunson, D. B. (2011). Bayesian spatial quantile regression. *J Am Stat Assoc*, 106(493):6–20.

Rousseeuw, P. J. and Leroy, A. M. (1987). *Robust Regression and Outlier Detection*. John Wiley and Sons, USA.

Sullivan, B. L., Aycrigg, J. L., Barry, J. H., et al. (2014). The ebird enterprise: An integrated approach to development and application of citizen science. *Biological Conservation*, 169:31–40.

Zhu, Z. (2017). Change detection using landsat time series: A review of frequencies, preprocessing, algorithms, and applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 130:370–384.