Modeling trend in temperature volatility using generalized LASSO

Anonymous Author(s)
Affiliation
Address
email

Abstract

words, words,

1 Introduction

Nonparametric variance estimation for spatio-temporal data.

1.1 Motivating applications

TODO: cut this down

There is a considerable interest in determining if there is an increasing trend in the climate variability [8, 10]. An increase in the temperature variability will increase the probability of extreme hot outliers. It might be harder for the society to adapt to these extremes than to the gradual increase in the mean temperature [10].

In this project, we consider the problem of detecting the trend in the temperature volatility. All analyses are performed on a sub-set of the European Centre for Medium-Range Weather Forecasts (ECMWF) ERA-40 dataset [26]. This dataset include the temperature measurements over a grid over the earth from 1957 to 2002. [6, 18, 19, 25, 27]

Research on analyzing the trend in the volatility of spatio-temporal data is scarce. [8] studied the change in the standard deviation (SD) of the surface temperature in the NASA Goddard Institute for Space Studies gridded temperature data set. In their analysis, for each geographical position, the mean of the temperature computed for the period 1951-1980 (called the base-period) at that position, is subtracted from the corresponding time series. Each time series is then divided by the standard deviation computed at each position and during the same time period. The distribution of the resulting data is then plotted for different periods. These distributions represent the deviation of the temperature for a specific period, from the mean in the base period, in units of the standard deviation in that period. The results showed that these distributions are widen for the resent time periods compared to 1951-1980. [10] took a similar approach in analysing the ERA-40 data set. However, in addition to the aforementioned method, they computed the distribution of the SDs in an alternative way: for each position and each time period, the deviation of the time-series at that position from the mean in that time period at that position was computed, and then divided by the SD of that position in the period before 1981. The results showed that there still is an increase in the SDs from 1958-1970 to 1991-2001, but this is much less than what is obtained from the method used in [8]. The authors also computed the time-evolving global SD from the de-trended time-series at each position. The resulting curve suggested that the global SD has been stable.

These previous work (and other related research, e.g., [16]) have several shortcomings. First, no statistical analysis has been performed to examine if the change in the SD is statistically significant.

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

Second, the methodologies for computing the SDs are rather arbitrary. The deviation of each time-series in a given period, is computed from either the mean of a base-period (as in [8]), or from the given period (as in [10, 16]). These deviations are then normalized using the SD of the base-period or the given period. No justification is provided for these choices. Third, the correlation between the observations is ignored. The observations in subsequent days and close geographical positions could be highly correlated. Without considering these correlations, any conclusion based on the averaged data could be flawed.

The main contribution of this work is to develop a new methodology for detecting the trend in the volatility of spatio-temporal data. In this methodology, the variance at each position and time, is considered as a hidden (un-observed) variable. The value of these hidden variables are then estimated by maximizing the likelihood of the observed data. We show that this formulation per se, is not appropriate for detecting the trend. To overcome this issue, we penalize the differences between the estimated variances of the observations which are temporally and/or spatially close to each other. This will result in an optimization problem called the *generalized LASSO problem* [21]. As we will see, the dimension of this optimization problem is very high and so the standard methods for solving the generalized LASSO cannot be applied directly. We investigate two methods for solving this optimization problem. In the first method, we adopt an optimization technique called alternative direction method of multipliers (ADMM) [4], to divide the total problem into several sub-problems of much lower dimension and show how the total problem can be solved by iteratively solving these sub-problems. The second method, called the *linearized ADMM algorithm* [14] solves the main problem by iteratively solving a linearized version of it. We will compare the benefits of each method.

Also neuroscience.

1.2 Related work

Mention [7, 12]. Also, [22, 23]. ARCH/GARCH.

[13, 17, 28] [15]

1.3 Main contributions

- We propose a model for non-parametric variance estimation for a spatio-temporal process (Section 2).
- We derive two algorithms to fit our estimator when applied to very large data (Section 3).
- We illustrate our methods on a large global temperature dataset with the goal of tracking world-wide trends in variance as well as a simulation constructed to mimic these data's features (Section 4).

2 ℓ_1-trend filtering for estimating variance of a time-series

ℓ_1-trend filtering was proposed by [11] as a method for estimating a smooth, time-varying trend. It is formulated as the optimization problem

$$\min_{\beta} \frac{1}{2}\sum_{t=1}^{T}(y_t-\beta_t)^2+\lambda\sum_{t=1}^{T-2}|\beta_t-2\beta_{t+1}+\beta_{t+2}|$$

or equivalently:

$$\min_{\beta} \frac{1}{2}\left\|y-\beta\right\|_2^2+\lambda\left\|D\beta\right\|_1 \tag{1}$$

where y_t is an observed time-series, β is the smooth trend, D is a $(T-2)\times T$ matrix, and λ is a tuning parameter which balances fidelity to the data (small errors in the first term) with a desire for smoothness. With the penalty matrix D, the estimated β will be piecewise linear. [11] proposed a specialized primal-dual interior point (PDIP) algorithm for solving (1). From a statistical perspective, (1) is a constrained maximum likelihood problem with independent observations from a normal distribution with common variance, $y_t \sim \mathrm{N}(\beta_t, \sigma^2)$, subject to a piecewise linear constraint on β.

Algorithm 1 PDIP for ℓ_1 variance estimation

```
Require: $\lambda > 0$, $w > 0$, $\nu \leftarrow 0$, $\mu_1 \leftarrow 0$, $\mu_2 \leftarrow 0$, $J \in \mathbb{Z}^+$, $\{w_k\}$    ▷ Initialization
  for $k = 1, 2, \ldots$ do                                                                    ▷ Central path
    for $j = 1, \ldots, J$ do                                                                 ▷ Newton updates
      Solve $A[\Delta\nu\ \Delta\mu_1\ \Delta\mu_2]^\top = r_{w_k}$ to find the search direction.
          $A$ is the Jacobian of $r_w$ in (4).
      Update $[\nu^{j+1}\ \mu_1^{j+1}\ \mu_2^{j+1}] \leftarrow [\nu^j\ \mu_1^j\ \mu_2^j] + [\Delta\nu\ \Delta\mu_1\ \Delta\mu_2]$.
    end for
  end for
  return $h = \log \frac{y^2}{1+D^\top\nu}$ TODO: Is this right? Explicit form of $A$
```

2.1 Estimating the variance

Inspired by the ℓ_1-trend filtering algorithm, we propose a non-parametric model for estimating the variance of a time-series. To this end, we assume that at each time step t, there is a hidden variable h_t such that conditioned on h_t the observations y_t are independent normal variables with zero mean and variance $\exp(h_t)$. The negative log-likelihood of the observed data in this model is $l(y \mid h) \propto -\sum_{t=1}^{T} h_t - y_t^2 e^{-h_t}$. Crucially, we assume that the hidden variables h_t vary smoothly. To impose this assumption, we estimate h_t by solving the penalized, negative log-likelihood:

$$\min_h -l(y \mid h) + \lambda \left\|Dh\right\|_1 \tag{2}$$

where D has the same structure as above.

TODO: explain the objective more. give the AR(1) example. Explain what you loses by this assumption (ACF,forecasting). Also explain that the covariace matrix is diagonal so it cannot capture the covariance structure. But incontrast to spatial stat literature, it does not make any assumption on estimated variances. Compare to Hallac et al and Lingren et al.

As with (1), one can solve (2) using the PDIP algorithm (as in, e.g., `cvxopt` [1]). First, we note that this is a generalized LASSO problem [21]. The dual of a generalized LASSO with the objective $f(x) + \lambda \left\|Dx\right\|_1$ is:

$$\min_\nu \quad f^*(-D^\top\nu) \qquad\qquad \text{s.t.} \quad \left\|\nu\right\|_\infty \leq \lambda$$

where $f^*(\cdot)$ is the Fenchel conjugate of f: $f^*(u) = \max_x u^\top x - f(x)$. It is simple to show that

$$f^*(u) = \sum_t (u_t - 1)\log\frac{y_t^2}{1-u_t} + u_t - 1. \tag{3}$$

Writing

$$r_w(v, \mu_1, \mu_2) := \begin{bmatrix} \nabla f^*(-D^\top v) + D(v - \lambda\mathbf{1})^\top\mu_1 - D(v + \lambda\mathbf{1})^\top\mu_2 \\ -\mu_1(v - \lambda\mathbf{1}) + \mu_2(v + \lambda\mathbf{1}) - w^{-1}\mathbf{1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4}$$

for $w > 0$, where μ_1 and μ_2 are dual variables for the ℓ_∞ constraint, as $w \to \infty$, the solution to this nonlinear system reduces to the KKT conditions. Therefore, the PDIP method takes Newton steps to solve the system for a series of increasing values of w. Algorithm 1 provides the details.

2.2 Adding spatial constraints

The method in the previous section can be used to estimate the variance of a single time-series. In this section, we extend this method to the estimation of the variance of spatio-temporal data.

At a specific time t, the data is measured on a grid of points with n_r rows and n_c columns for a total of $S = n_r \times n_c$ spatial locations. Let y_{ijt} denote the value of the observation at time t on the i^{th} row and j^{th} column of the grid, and h_{ijt} denote the corresponding hidden variable. We seek to impose both temporal and spatial smoothness constraints on the hidden variables. Specifically, we seek a solution for h which is piecewise linear in time and piecewise constant in space (although higher-order smoothness can be imposed with minimal alterations to the methodology). We achieve this goal by solving the following optimization problem:

$$\min_h \sum_{i,j,t} h_{ijt} + y_{ijt}^2 e^{-h_{ijt}} + \lambda_1 \sum_{i,j} \sum_{t=1}^{T-2} \left| h_{ijt} - 2h_{ij(t+1)} + h_{ij(t+2)} \right|$$
$$+ \lambda_2 \sum_{t,j} \sum_{i=1}^{n_r-1} \left| h_{ijt} - h_{(i+1)jt} \right| + \lambda_2 \sum_{t,i} \sum_{j=1}^{n_c-1} \left| h_{ijt} - h_{i(j+1)t} \right| \tag{5}$$

The first term in the objective is proportional to the negative log-likelihood, the second is the temporal penalty for the time-series at each location (i, j), while the third and fourth, penalize the difference between the estimated variance of two vertically and horizontally adjacent points, respectively. The spatial component of this penalty is a special case of trend filtering on graphs [29] which penalizes the difference between the estimated values of the signal on the connected nodes. As before, we can write (5) in matrix form where h is an $T \times S$ vector and D is replaced by D_{TS}. Then, as we have two different tuning parameters for the temporal and spatial components, we write $\lambda = \left[\lambda_1 \mathbf{1}_T^\top,\ \lambda_2 \mathbf{1}_S^\top\right]^\top$ and $\Lambda = \mathtt{diag}(\lambda)$ leading to

$$\min_h -l(y \mid h) + \|\Lambda D_{TS} h\|_1 . \tag{6}$$

The dual of this problem is $\min_\nu f^*(-D_{ST}^\top \nu)$ subject to $|\nu_k| \leq \Lambda_{kk},\ k = 1, \ldots, ST$, where f^* is given in (3).

TODO: Add back the full form of D above? It was too cumbersome before, but possibly necessary

3 Proposed optimization methods

For a spatial grid of size S and T time steps, D_{ST} will have $3n_r n_c + Tn_c + 2Tn_r$ rows and ST columns. For a $1° \times 1°$ grid over the entire northern hemisphere and daily data over 10 years, we have $n_r = 90$, $n_c = 180$, $T = 3650$ and so D_{ST} has approximately 10^8 columns and 10^8 rows. In each step of Algorithm 1, we need to solve a linear system of equations in A which depends on $D_{ST}^\top D_{ST}$ (see [3] equation 11.54). Therefore, applying the PDIP directly is infeasible for our data.[1]

In the next section, we develop two ADMM algorithms for solving this problem efficiently. The first casts the problem as a so-called consensus optimization problem [4] which solves smaller sub-problems using PDIP and then recombines the results. The second uses proximal methods to avoid matrix inversions. TODO: We note that stochastic gradient descent could be used but...

3.1 Consensus optimization

General (unconstrained) optimization problems $\min_z \sum_i f_i(z)$, $z \in \mathbb{R}^n$ can be rewritten equivalently using a collection of local variables, $x_i \in \mathbb{R}^{n_i}$, and a constraint. Following the notation of [4], let $k = \mathscr{G}(i, j)$ which means that the j^{th} entry of x_i is z_k (or $(x_i)_j = z_k$) and define $\tilde{z}_i \in \mathbb{R}^{n_i}$ by $(\tilde{z}_i)_j = (x_i)_j$. Then the original unconstrained optimization problem is equivalent to the following constrained problem $\min_{\{x_1,\ldots,x_N\}} \sum_i f_i(x_i)$ subject to $\tilde{z}_i = x_i$. Now, we can apply ADMM to the augmented Lagrangian of this problem:

$$\begin{aligned}
x_i &\leftarrow \operatorname*{argmin}_{x_i} f_i(x_i) + (u_i)^\top x_i + (\rho/2)\, \|x_i - \tilde{z}_i\|_2^2 \\
z_k &\leftarrow (1/S_k) \sum_{\mathscr{G}(i,j)=k} (x_i)_j \\
u_i &\leftarrow u_i + \rho(x_i - \tilde{z}_i)
\end{aligned} \tag{7}$$

Here, S_k is the number of local variable entries that correspond to z_k, and u_i are the Lagrange multipliers. The x-update is often a simple modification as the original optimization, though over only a small partition, the z-update simply averages over collections of analogous local variables

[1] We note that this is a highly structured and sparse matrix, but, unlike trend filtering alone, it is not banded. We are unaware of general linear algebra techniques for inverting such matrix, despite our best efforts.

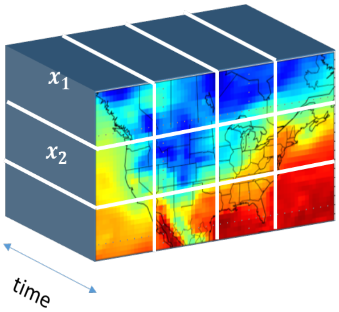

Figure 1: The cube represents the global variable z in space and time. The sub-cubes specified by the white lines are x_i. TODO: Can we use h_i here instead of x_i?

while the u update enforces sharing. Both the x and u updates are parallelizable over the partition of local variables. TODO: I don't much like this explanation or notation. Seems unclear

To solve the optimization problem (6) or (5) using this method, we need to two address two questions: first, how to choose the local variables x_i, and second, how to solve the optimization problem for updating these variables (the first line of (7)).

In Figure 1, the global variable z is represented as a cube (using the subset of the US as an example). We decompose z into sub-cubes as shown in the figure by white lines. It is easy to see that with this definition of x_i, the objective (6) decomposes as $\sum_i f_i(x_i)$ where $f_i(x_i) = -l(y_i \mid x_i) + \left\|\Lambda_{(i)} D_{(i)} x_i\right\|$, and $\Lambda_{(i)}$ and $D_{(i)}$ contain the temporal and spatial penalties corresponding to x_i only. TODO: I would like to be able to write all this in terms of h instead of x

By this definition of x_i, the update step for x_i is the following optimization problem: $x_i^{m+1} := \text{argmin}_{x_i}(f_i(x_i) + (u_i^m)^t x_i + (\rho/2)\|x_i - \hat{z}_i^m\|_2^2)$.

We solve this using the PDIP method. To this end we first need to compute the dual of the problem. It can be shown that the dual of this problem is:

$$\min_{\nu} f_i^*(-D_i^t \nu_i) \qquad\qquad s.t. |(\nu_i)_j| \le (\Lambda_i)_j \tag{8}$$

where:

$$\begin{aligned} f_i^*(q) &= \sum_j (q_i)_j (w_i)_j - (w_i)_j - (y_i)_j^2 e^{-(w_i)_j} - (\rho/2)((w_i)_j - (\alpha_i^m)_j)^2 \\ \alpha_i^m &= \hat{z}_i^m - u_i^m \\ (w_i)_j &= \mathscr{W}\left(\frac{(y_i)_j^2}{\rho} \exp\left[\frac{1 - q_j - \rho(\alpha_i^m)_j}{\rho}\right]\right) - \frac{1 - q_j - \rho(\alpha_i^m)_j}{\rho} \end{aligned} \tag{9}$$

TODO: I'm not quite sure what's going on here. These need to match the generic discussion in (7). Or are these just defining some other variables for convenience? This needs to be clarified and incorporated into Algorithm 2.

In these equations, $\mathscr{W}$ is the *Lambert function* [5]. To use the PDIP method, we also need to compute the gradient and Hessian of $f_i^*(-D_i^t \nu_i)$. This involves computing the derivatives of the Lambert functions. The primal solution x_i can be obtained from the dual solution ν_i in (8) by setting $x_i = w_i$ where w_i is defined in the last equation of (9).

The complete ADMM algorithm for estimating the variances is represented in Algorithm 2. All the computations in the three updating steps (7) can be performed in parallel. The number of rows and columns of the sub-cubes should be chosen so that the updating of x_i could be performed in one processor. We choose 3 × 3 × 521 sub-cubes.

Because Algorithm 2 breaks the large optimization into sub-problems that can be solved independently, it is amenable to a split-gather parallelization strategy via, e.g., the map reduce framework. In each iteration, the computation time will be equal to the time to solve each sub-problem plus the time to communicate the solutions on the master processor and perform the consensus step. Since each

Algorithm 2 ADMM for sparse estimation of variance of spatio-temporal data TODO: Fix this

```
Input: data $y$, mapping $\mathscr{G}(i,j)$, $\rho, \lambda_t, \lambda_s$
Initialization: $x_i^0 = z^0 = u_i^0 = \mathbf{0}$.
for $m = 1, 2, \ldots$ do
    for $i = 1$ to $N_{sub-cubes}$ do
        compute $\nu_i$ from (8)
        compute $w_i$ from (9)
        set $x_i^m := w_i$
    end for
    Compute $z^m$ from (7)
    Compute $u_i^m$ from (7)
end for
```

sub-problem is small, with parallelization, the computation time in each iteration will be small. In addition, our experiments with several values of λ_t and λ_s showed that the algorithm converges in few hundreds iterations. TODO: Need to redo this:Solving each sub-problem on a machine with four 3.20GHz Intel i5-3470 cores takes less than 3 seconds on average, and so for example if we assume that communication time is 10 seconds and the algorithm converges in 300 iterations, with parallelization on $N_{sub-cubes}$ machines, the algorithm will converge in about 1 hour. Assuming that we use $N_{sub-cubes}$ machines and that the convergence rate of the algorithm is independent of the grid size, this time will be independent of the grid size.

If we perform these computations on a single machine, the computation time grows linearly with $N_{sub-cubes}$. For example, for the data in a grid over the united states and using $3 \times 3 \times 521$ sub-cubes each iteration of the algorithm will take about 20 minutes on a single machine and so with 300 iterations it will take several days to converge. Given that we need to compute the solution for several values of the parameters λ_t and λ_s, this computation time is not feasible.

Therefore, this algorithm is only useful if we can parallelize the computation over several machines. In the next section, we describe another algorithm which makes the computation feasible on a single machine.

3.2 Linearized ADMM

TODO: Need a good dummy variable here, it can't be u. Consider the generic optimization problem $\min_x f(x) + g(Dx)$ where $x \in \mathbb{R}^n$ and $D \in \mathbb{R}^{m \times n}$. Each iteration of the linearized ADMM algorithm [14] for solving this problem has the form

$$\begin{aligned} x &\leftarrow \underset{\mu f}{\mathbf{prox}} \left(x - (\mu/\rho) D^T (Dx - z + u) \right) \\ z &\leftarrow \underset{\rho g}{\mathbf{prox}} \left(z + u \right) \\ u &\leftarrow u + Dx - z \end{aligned} \tag{10}$$

where $z, u \in \mathbb{R}^m$ and the proximal operator is defined as

$$\underset{\alpha f}{\mathbf{prox}}(u) = \min_x \ \alpha \cdot f(x) + \frac{1}{2} \|x - u\|_2^2 \, .$$

Clearly, (5) has this form.TODO: What are μ and ρ? To perform the steps in (10), we need to evaluate $\mathbf{prox}_{\mu f}$ and $\mathbf{prox}_{\rho g}$. Proximal algorithms are feasible only if these proximal operators can be evaluated efficiently which, as we show next, is the case for our problem.

Theorem 1. *Let $f(h) = \sum_k h_k + y_k^2 e^{-h_k}$ and $g(x) = \|x\|_1$. Then,*

$$\begin{aligned} [\underset{\mu f}{\mathbf{prox}}(u)]_k &= \mathscr{W}\left(\frac{y_k^2}{\mu} \exp\left(\frac{1 - \mu u_k}{\mu} \right) \right) + \frac{1 - \mu u_k}{\mu}, \\ \underset{\rho g}{\mathbf{prox}}(u) &= S_{\rho\lambda}(u) \end{aligned}$$

where $[S_\alpha(u)]_k = \operatorname{sign}(u_k)(|u_k| - \alpha_k)_+$ and $(v)_+ = v \vee 0$.

Algorithm 3 Linearized ADMM

```
Input: data $y$, penalty matrix $D$, $\rho, \lambda_t, \lambda_s > 0$.
Set: $h \leftarrow 0$, $z \leftarrow 0$, $u \leftarrow 0$.
for $m = 1, 2, \ldots$ do
    $h_k \leftarrow \mathscr{W}\left(\frac{y_k^2}{\mu} \exp\left(\frac{1-\mu u_k}{\mu}\right)\right) + \frac{1-\mu u_k}{\mu}$,
    $z \leftarrow S_{\rho\lambda}(u)$.
    $u \leftarrow u + Dh - z$
end for
```

Table 1: Parameters used to simulate data. TODO: Any ideas to take up less space with this info?

s	r_s	c_s	σ_s	α_s	ω_s	ϕ_s
1	0	0	5	0.5	0.121	0
2	0	5	5	0.1	0.121	0
3	3	0	5	-0.5	0.121	$\pi/2$
4	3	5	5	-0.1	0.121	$\pi/2$

Proof. If $f(x) = \sum_k f_k(x_k)$ then $[\mathbf{prox}_{\mu f}(x)]_k = \mathbf{prox}_{\mu f_k}(u_k)$. So $[\mathbf{prox}_{\mu f}(u)]_k = \min_{x_k} \; \mu\big(x_k + y_k^2 e^{-x_k}\big) + \frac{1}{2}(x_k - u_k)^2$. Setting the derivative to 0 and solving for u_k gives the result. Similarly, $[\mathbf{prox}_{\rho g}(u)]_\ell = \rho\lambda_\ell |z_\ell| + 1/2(z_\ell - u_\ell)^2$. This is not differentiable, but the solution must satisfy $\rho \cdot \lambda_\ell \cdot \partial\big(|z_\ell|\big) = u_\ell - z_\ell$ where $\partial\big(|z_\ell|\big)$ is the sub-differential of $|z_\ell|$. The solution is the soft-thresholding operator $S_{\rho\lambda_\ell}(u_\ell)$. □

4 Empirical evaluation

In this section, we examine both simulated and real spatio-temporal climate data. All the computations were performed on a Linux machine with four 3.20GHz Intel i5-3470 cores.

4.1 Simulations

TODO: Can we add some type of performance measure? What if we fit 2 marginal models, spatial only and temporal only. Plus maybe a marginal GARCH?

We generate observations at all time steps and all locations from independent Gaussian random variables with zero mean. However, the variance of these random variables follows a smoothly varying function in time and space

$$\sigma^2(t, r, c) = \sum_{s=1}^{S} W_s(t) \cdot \exp\left(\frac{(r - r_s)^2 + (c - c_s)^2}{2\sigma_s^2}\right); \quad W_s(t) = \alpha_s \cdot t + \exp(\sin(2\pi\omega_s t + \phi_s)).$$

In words, the variance at each time and location is computed as the weighted sum of S bell-shaped functions where the weights are time-varying, consist of a linear trend $\alpha_s \cdot t$ and a periodic term $\beta_s \cdot \sin(2\pi\omega_s t + \phi_s)$. The bell-shaped functions impose the spatial smoothness, and the linear trend and the periodic terms enforce the temporal smoothness similar to the seasonal component in the real climate data. We simulated the data on a 5 by 7 grid and for 780 time steps with $S = 4$. The parameters of the variance function are shown in Table 1. For reference, we plot the variance function for all locations at $t = 25$ and $t = 45$ in as well as the variance across time at $(0, 0)$ in Figure 2.

We estimated the linearized ADMM for all combinations of values of λ_t and λ_s from the sets $\lambda_t \in \{0, 1, 5, 10, 50, 100\}$ and $\lambda_s \in \{0, 0.05, 0.1, 0.2, 0.3\}$. For each pair, we then compute the mean absolute error (MAE) between the estimated variance and the true variance at all locations and all time steps. For $\lambda_t = 5$ and $\lambda_s = 0.1$ MAE was minimized. The right panel of Figure 3 shows the true and the estimated standard deviation at location (0,0) using $\lambda_s = 0.1$ and $\lambda_t = 5$ (blue) and

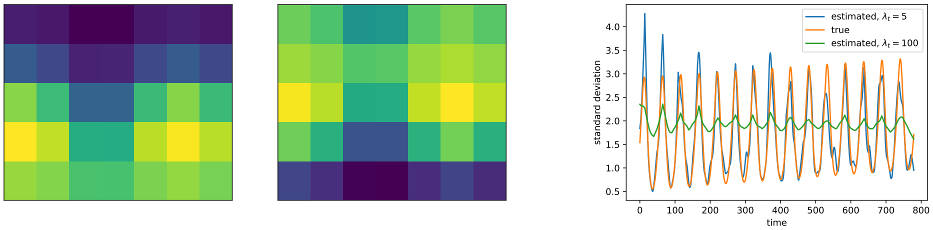

Figure 2: Variance function at $t = 25$ (left) and $t = 45$ (center). Right: the true (orange) and estimated standard deviation function at the location (0,0). The estimated values are obtained using linearized ADMM with $\lambda_s = 0.1$ and two values of λ_t: $\lambda_t = 5$ (blue) and $\lambda_t = 100$ (green).

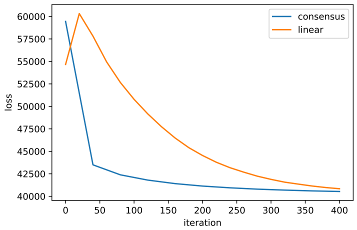

Figure 3: Convergence speed of linearized and consensus ADMM.

$\lambda_t = 100$ (green). As we can see, larger than optimal value of λ_t leads to estimated values which are “too smooth".

Figure 3 shows the convergence of Algorithms 2 and 3. Each iteration of the linearized algorithm takes 0.01 seconds on average while each iteration of the consensus ADMM takes about 20 seconds. TODO: These data are pretty small. Is it possible to do the PDIP?

4.2 Data analysis

Algorithm 2 is appropriate only if we parallelize it over multiple machines, and it is significantly slower on our simulated data, so we do not pursue it further here. All the results reported in this section are obtained using Algorithm 3. We applied this algorithm to the northern hemisphere of the ERA-40 dataset available from the https://www.ecmwf.int. The data are the 2 meter temperature measured daily at 12 p.m from August 31 of 1992 to 2002.

TODO: From here down needs to be cut down and corrected with the right numbers.

Data Exploration The red curve in the left panel of Figure 4 shows the estimated SD (which is $\exp(h_t/2)$) of the residuals of the time-series of Bloomington. To reduce the number of time-steps in this figure and in the remainder of the paper we work on the weekly averaged of the data.

The curve of the estimated SD captures the periodic variations in the SD of the signal. Just by looking at this curve, it is hard to say if the SD is decreasing or increasing. Therefore, we compute the average of the estimated SD for each year. The estimated SD together with this annual average is shown in the middle panel of Figure 4. As it can be seen, the annual trend is not smooth. This is because in the optimization problem (2), the smoothness of the annual trend is not encouraged. To remedy this, we add the following long horizon penalty to (2):

$$\sum_{i=1}^{N_{year}-2} \left| \sum_{t=1}^{52} h_{t_1} - 2h_{t_2} + h_{t_3} \right| \tag{11}$$

where $t_1 = 52(i-1)+t$, $t_2 = 52i+t$ and $t_3 = 52(i+1)+t$. Also, N_{year} is the number of years over which we are performing our analysis (here $N_{year} = 10$). Since we are working on the weekly averaged data, each year corresponds to 52 observations. In the matrix form, the penalty (11) adds

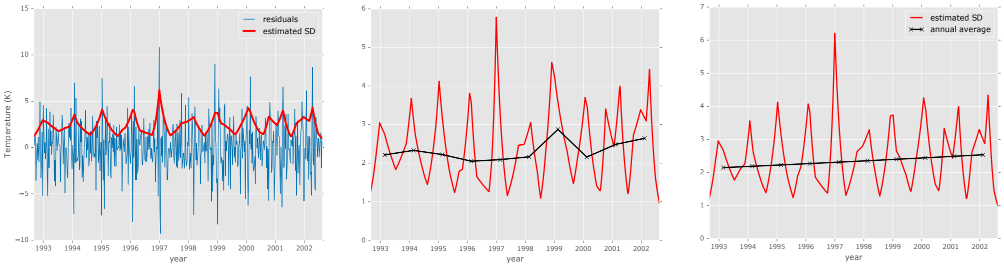

Figure 4: Left: The residuals of the time-series of Bloomington (averaged weekly) and the estimated SD obtained from the method of Section 2.1 (red). Middle: the estimated SDs (red) and their annual average (black) without imposing the long horizon penalty. Right: the same as middle panel but here the long horizon penalty is imposed. See the text for more details.

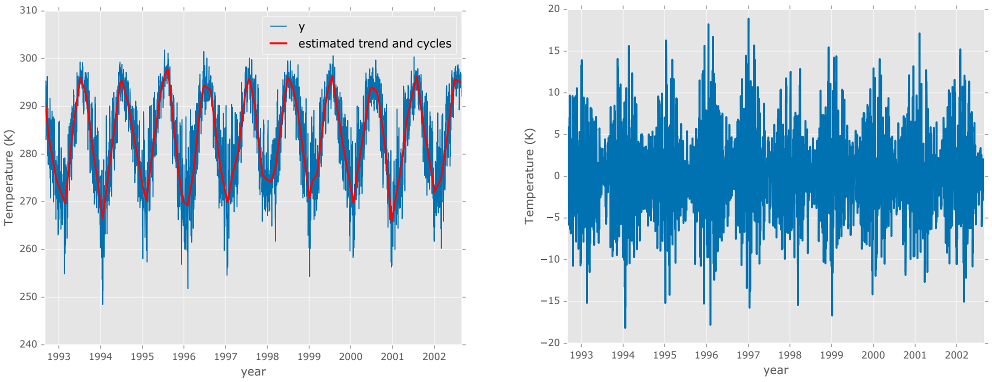

Figure 5: Left: Time-series of the temperature of Bloomington (blue) and the estimated trend and cycles obtained from the ℓ_1-trend filtering (red). Right: the same time-series after removing the cyclic terms and de-trending using ℓ_1-trend filtering.

N_{year} rows to the matrix D. The estimated SDs using this penalty matrix is shown in the right panel of Figure 4. The annual average of the estimated SDs shows a linear trend with a positive slope.

This section is devoted to exploring some of the properties of the ERA-40 surface temperature data set. The goal here is to demonstrate some of the difficulties in modeling the trend in the temperature volatility and motivate our methodology for doing so.

The right panel of Figure 5 shows the time-series of the temperature of Bloomington, after removing the cyclic terms and de-trending using the method explained in the next section. The goal is to investigate the trend in the variance of this signal. This figure, reveals another issue toward this goal: the variance of this signal, shows cyclic behavior. Also, the cycles are not regular and their amplitude and frequency change. Even if one can describe the behavior of the variance of all the time-series using a single parametric model (for example a variant of the GARCH models [2]), it is not clear how the trend in the variance should be investigated in this framework. These observations motivate the need to develop a non-parametric framework for the problem at hand.

Convergence We used the following rule to determine when to stop the optimization: the optimization was stopped if the value of the loss did not improve by at least 0.1% in 1000 trials. As we can see, the algorithm converged in about 2000 iterations. This took about 11 minutes. Our experiments showed that the convergence speed depends on the value of λ_t and λ_s. Also, if we use the solution obtained for smaller values of these parameters as the initial value for the larger values (*warm start*), the converges speed improves.

Model selection One common method for choosing the penalty parameters in the Lasso problems is to find the solution for a range of the values of these parameters and then choose the values which minimize a model selection criterion. However, such analyses needs the computation of the degrees of freedom (df). Several previous work have investigated the df in generalized lasso problems [9, 24, 30]. However, all these studies have considered the linear regression problem and, to the best of our knowledge, the problem of computing the df for generalized lasso with general objective function has not been considered yet.

Another approach is to choose the set of values which minimize an estimate of the expected prediction error obtained by k-fold cross-validation [20] . Although this method is applicable for our problem, it needs k times more computation.

In this paper, we use a heuristic method for choosing λ_t and λ_s: we compute the optimal solution for a range of values of these parameters and choose the values which minimize $\mathscr{L}(\lambda_t, \lambda_s) = -l(y|h)+\sum\|D_{total}h\|$. This objective is a compromise between the negative log likelihood ($-l(y|h)$) and the complexity of the solution ($\sum\|D_{total}h\|$). For smoother solutions the value of $\sum\|D_{total}h\|$ will be smaller but with the cost of larger $-l(y|h$.

We computed the optimal solution for all the combinations of the following sets of values: $\lambda_t \in \{1, 5, 10, 20\}$, $\lambda_s \in \{0, .1, 1, 5, 10\}$. The best combination based on a held out set was $\lambda_t = 5$ and $\lambda_s = 1$. All the analyses in the next section are performed on the solution for these values.

Analysis of trend of temperature volatility The top row of Figure 6 shows the detrended data, the estimated standard deviation and the yearly average of these estimates for two cities in the US. The estimated SD captures the periodic behavior in the variance of the time-series. In addition, the number of linear segments changes adaptively in each time window depending on how fast the variance is changing.

The yearly average of the estimated SD captures the trend in the temperature volatility. For example, we can see that in Bloomington, there is a small positive trend. To determine how the volatility has changed in each location, we subtract the average of the estimated variance in 1992 from the average in the following years and compute their sum. The value of this change in the variance in each location is depicted in the right panel of Figure 6. The left panel of this figure, shows the average estimated variance in each location.

It is interesting to note that the trend in volatility is almost zero over the oceans. The most positive trend can be observed in the south-east and the most negative trend has happened in the north-east.

5 Discussion

In this paper, we proposed a new method for estimating the variance of spatio-temporal data. The main idea is to cast this problem as a constrained optimization problem where the constraints enforce smooth changes in the variance for neighboring points in time and space. In particular, the solution is piecewise linear in time and piecewise constant in space. The resulting optimization is in the form of a generalized LASSO problem with high-dimension, and so applying the PDIP method directly is infeasible. We therefore developed two ADMM-based algorithms to solve this problem: the consensus ADMM and linearized ADMM.

The consensus ADMM algorithm converges in few hundreds of iterations but each iteration takes much longer than the linearized ADMM algorithm. The appealing feature of the consensus ADMM algorithm is that if it is parallelized on enough number of machines the computation time per iteration remains constant as the problem size increases. The linearized ADMM algorithm, on the other hand converges in few thousands of iterations but each iteration is performed in split second. However, since the algorithm converges in many iterations it is not very appropriate for parallelization. The reason is that after each iteration the solution computed in each machine should be broadcast to the master machine and this operation takes some time which depends on the speed of the network connecting the slave machines to the master. A direction for future research would be to combine

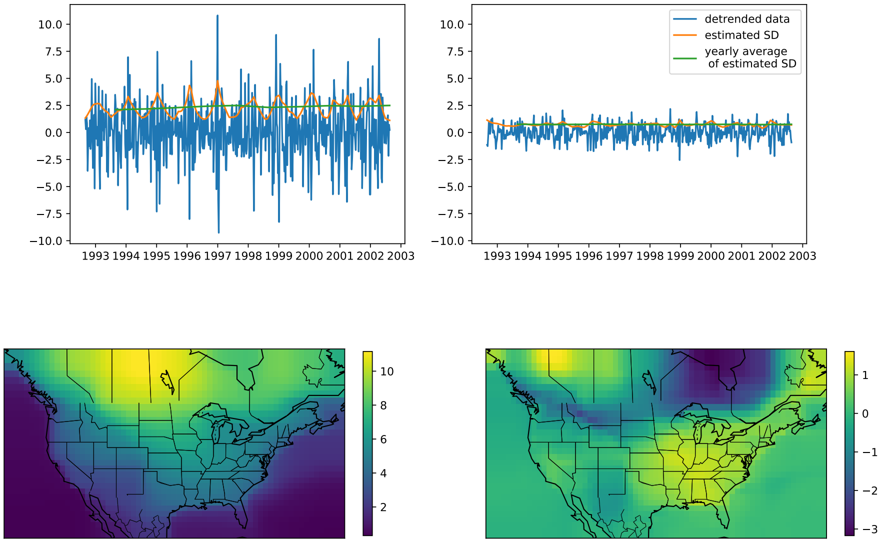

Figure 6: Top row: Detrended data and the estimated SD for Bloomington (left) and San Diego (right). Bottom: the average of the estimated variance over the US (left) and the change in the variance from 1992 to 2002 (right)

these two algorithms in the following way: the problem should be split into the sub-problems (as in the consensus ADMM) but each sub-problem can be solved using linearized ADMM.

We applied the linearized ADMM algorithm to the surface temperature data on a grid over the united states, for years 1992-2002. The results showed that in many locations the variance of the temperature has increased about 1 unit in 10 years.

The goal of this paper, however, is not to make any conclusions about the trend in the variance because we solved the problem only for a grid over the united states and for 10 years of the data. A thorough analysis, needs the full solution over the globe and for a longer time period. The goal of the paper, was to propose the idea of estimating the trend in variance of spatio-temporal signals using generalized lasso and to investigate the algorithms for solving the resulting optimization problem.

References

[1] M. S. Andersen, J. Dahl, and L. Vandenberghe. CVXOPT: A Python package for convex optimization, version 1.1. 6. *Available at cvxopt. org 54*, 2013.

[2] T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3): 307–327, Apr. 1986. ISSN 0304-4076.

[3] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, 3(1): 1–122, 2011. ISSN 1935-8237.

[5] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth. On the LambertW function. *Advances in Computational Mathematics*, 5(1):329–359, Dec. 1996.

[6] E. M. Fischer, U. Beyerle, and R. Knutti. Robust spatially aggregated projections of climate extremes. *Nature Climate Change*, 3:1033—1038, 2013. URL `http://dx.doi.org/10.1038/nclimate2051`.

[7] D. Hallac, Y. Park, S. Boyd, and J. Leskovec. Network inference via the time-varying graphical lasso. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data*

Mining, KDD '17, pages 205–213, New York, NY, USA, 2017. ACM. doi: 10.1145/3097983.3098037. URL http://doi.acm.org/10.1145/3097983.3098037.

[8] J. Hansen, M. Sato, and R. Ruedy. Perception of climate change. *Proceedings of the National Academy of Sciences*, 109(37), Sept. 2012.

[9] Q. Hu, P. Zeng, and L. Lin. The dual and degrees of freedom of linearly constrained generalized lasso. *Computational Statistics & Data Analysis*, 86:13–26, June 2015.

[10] C. Huntingford, P. D. Jones, V. N. Livina, T. M. Lenton, and P. M. Cox. No increase in global temperature variability despite changing regional patterns. *Nature*, 500(7462):327–330, Aug. 2013. ISSN 0028-0836.

[11] S. Kim, K. Koh, S. Boyd, and D. Gorinevsky. ℓ_1 Trend Filtering. *SIAM Review*, 51(2):339–360, May 2009. ISSN 0036-1445. doi: 10.1137/070690274. URL http://epubs.siam.org/doi/abs/10.1137/070690274.

[12] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky. ℓ_1 trend filtering. *SIAM Review*, 51(2):339–360, 2009. doi: 10.1137/070690274. URL https://doi.org/10.1137/070690274.

[13] K. Lin, J. L. Sharpnack, A. Rinaldo, and R. J. Tibshirani. A sharp error analysis for the fused lasso, with application to approximate changepoint screening. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6884–6893. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7264-a-sharp-error-analysis-for-the-fused-lasso-with-application-to-approximate-changepoint-screening.pdf.

[14] N. Parikh and S. Boyd. Proximal Algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, Jan. 2014.

[15] A. Ramdas and R. J. Tibshirani. Fast and flexible admm algorithms for trend filtering. *Journal of Computational and Graphical Statistics*, 25(3):839–858, 2016.

[16] A. Rhines and P. Huybers. Frequent summer temperature extremes reflect changes in the mean, not the variance. *Proceedings of the National Academy of Sciences*, 110(7):E546–E546, Feb. 2013.

[17] V. Sadhanala, Y.-X. Wang, J. L. Sharpnack, and R. J. Tibshirani. Higher-order total variation classes on grids: Minimax theory and trend filtering methods. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5800–5810. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7162-higher-order-total-variation-classes-on-grids-minimax-theory-and-trend-filtering-methods.pdf.

[18] J. A. Screen. Arctic amplification decreases temperature variance in northern mid- to high-latitudes. *Nature Climate Change*, 4:577—582, 2014. URL http://dx.doi.org/10.1038/nclimate2268.

[19] P. W. Staten, B. H. Kahn, M. M. Schreier, and A. K. Heidinger. Subpixel characterization of HIRS spectral radiances using cloud properties from AVHRR. *Journal of Atmospheric and Oceanic Technology*, 33(7):1519–1538, 2016. doi: 10.1175/JTECH-D-15-0187.1.

[20] R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[21] R. J. Tibshirani. *The Solution Path of the Generalized Lasso*. PhD Thesis, Stanford University, 2011.

[22] R. J. Tibshirani. Adaptive piecewise polynomial estimation via trend filtering. *Annals of Statistics*, 42:285–323, 2014. URL http://www.stat.cmu.edu/~ryantibs/papers/trendfilter.pdf.

[23] R. J. Tibshirani and J. Taylor. The solution path of the generalized lasso. *Annals of Statistics*, 39(3):1335–1371, 2011.

[24] R. J. Tibshirani and J. Taylor. Degrees of freedom in lasso problems. *The Annals of Statistics*, 40(2):1198–1232, 2012.

[25] K. E. Trenberth, Y. Zhang, J. T. Fasullo, and S. Taguchi. Climate variability and relationships between top-of-atmosphere radiation and temperatures on earth. *Journal of Geophysical Research: Atmospheres*, 120(9):3642–3659, 2014. doi: 10.1002/2014JD022887.

[26] S. M. Uppala, P. W. KÅllberg, A. J. Simmons, U. Andrae, and e. al. The ERA-40 re-analysis. *Quarterly Journal of the Royal Meteorological Society*, 131(612):2961–3012, Oct. 2005.

[27] D. A. Vasseur, J. P. DeLong, B. Gilbert, H. S. Greig, C. D. G. Harley, K. S. McCann, V. Savage, T. D. Tunney, and M. I. O'Connor. Increased temperature variation poses a greater risk to species than climate warming. *Proceedings of the Royal Society of London B: Biological Sciences*, 281(1779), 2014. doi: 10.1098/rspb.2013.2612.

[28] Y.-X. Wang, J. Sharpnack, A. J. Smola, and R. J. Tibshirani. Trend filtering on graphs. *Journal of Machine Learning Research*, 17(105):1–41, 2016. URL `http://jmlr.org/papers/v17/15-147.html`.

[29] Y.-X. Wang, J. Sharpnack, A. J. Smola, and R. J. Tibshirani. Trend Filtering on Graphs. *Journal of Machine Learning Research*, 17(105):1–41, 2016. URL `http://jmlr.org/papers/v17/15-147.html`.

[30] P. Zeng, Q. Hu, and X. Li. Geometry and Degrees of Freedom of Linearly Constrained Generalized Lasso. *Scandinavian Journal of Statistics*, 44(4):989–1008, Nov. 2017. ISSN 0303-6898.