
Modeling trend in temperature volatility using generalized LASSO

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 words, words,

2 1 Introduction

3 **TODO: Some equations do not have numbering and some have.**

4 Nonparametric variance estimation for spatio-temporal data.

5 1.1 Motivating applications

6 **TODO: cut this down**

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

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

15 Research on analyzing the trend in the volatility of spatio-temporal data is scarce. [8] studied the
16 change in the standard deviation (SD) of the surface temperature in the NASA Goddard Institute
17 for Space Studies gridded temperature data set. In their analysis, for each geographical position,
18 the mean of the temperature computed for the period 1951-1980 (called the base-period) at that
19 position, is subtracted from the corresponding time series. Each time series is then divided by the
20 standard deviation computed at each position and during the same time period. The distribution of
21 the resulting data is then plotted for different periods. These distributions represent the deviation
22 of the temperature for a specific period, from the mean in the base period, in units of the standard
23 deviation in that period. The results showed that these distributions are widen for the resent time
24 periods compared to 1951-1980. [10] took a similar approach in analysing the ERA-40 data set.
25 However, in addition to the aforementioned method, they computed the distribution of the SDs in
26 an alternative way: for each position and each time period, the deviation of the time-series at that
27 position from the mean in that time period at that position was computed, and then divided by the SD
28 of that position in the period before 1981. The results showed that there still is an increase in the SDs
29 from 1958-1970 to 1991-2001, but this is much less than what is obtained from the method used in
30 [8]. The authors also computed the time-evolving global SD from the de-trended time-series at each
31 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. 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

TODO: Arash: shouldn't the title of this section be "...estimating variance of spatio-temporal data" or something similar?

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:

$$\min_{\beta} \frac{1}{2} \|y - \beta\|_2^2 + \lambda \|D\beta\|_1 \quad (1)$$

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,

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 $k = 1, 2, \dots$ **do** ▷ Central path
 for $j = 1, \dots, J$ **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 (3).
 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

76 (1) is a constrained maximum likelihood problem with independent observations from a normal
77 distribution with common variance, $y_t \sim \mathcal{N}(\beta_t, \sigma^2)$, subject to a piecewise linear constraint on β .

78 2.1 Estimating the variance

79 Inspired by the ℓ_1 -trend filtering algorithm, we propose a non-parametric model for estimating
80 the variance of a time-series. To this end, we assume that at each time step t , there is a hidden
81 variable h_t such that conditioned on h_t the observations y_t are independent normal variables with
82 zero mean and variance $\exp(h_t)$. The negative log-likelihood of the observed data in this model is
83 $l(y | h) \propto -\sum_{t=1}^T h_t - y_t^2 e^{-h_t}$. Crucially, we assume that the hidden variables h_t vary smoothly.
84 To impose this assumption, we estimate h_t by solving the penalized, negative log-likelihood:

$$\min_h -l(y | h) + \lambda \|Dh\|_1 \quad (2)$$

85 where D has the same structure as above.

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

90 As with (1), one can solve (2) using the PDIP algorithm (as in, e.g., cvxopt [1]). First, we note that
91 this is a generalized LASSO problem [21]. The dual of a generalized LASSO with the objective
92 $f(x) + \lambda \|Dx\|_1$ is:

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

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

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

94 Writing

$$r_w(v, \mu_1, \mu_2) := \begin{bmatrix} \nabla f^*(-D^\top v) + D(v - \lambda \mathbf{1})^\top \mu_1 - D(v + \lambda \mathbf{1})^\top \mu_2 \\ -\mu_1(v - \lambda \mathbf{1}) + \mu_2(v + \lambda \mathbf{1}) - w^{-1} \mathbf{1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (3)$$

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

98 2.2 Adding spatial constraints

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

101 At a specific time t , the data is measured on a grid of points with n_r rows and n_c columns for a
102 total of $S = n_r \times n_c$ spatial locations. Let y_{ijt} denote the value of the observation at time t on the

103 i^{th} row and j^{th} column of the grid, and h_{ijt} denote the corresponding hidden variable. We seek to
 104 impose both temporal and spatial smoothness constraints on the hidden variables. Specifically, we
 105 seek a solution for h which is piecewise linear in time and piecewise constant in space (although
 106 higher-order smoothness can be imposed with minimal alterations to the methodology). We achieve
 107 this goal by solving the following optimization problem:

$$\begin{aligned} \min_h \sum_{i,j,t} h_{ijt} + y_{ijt}^2 e^{-h_{ijt}} + \lambda_1 \sum_{i,j} \sum_{t=1}^{T-2} |h_{ijt} - 2h_{ij(t+1)} + h_{ij(t+2)}| \\ + \lambda_2 \sum_{t,j} \sum_{i=1}^{n_r-1} |h_{ijt} - h_{(i+1)jt}| + \lambda_2 \sum_{t,i} \sum_{j=1}^{n_c-1} |h_{ijt} - h_{i(j+1)t}| \end{aligned} \quad (4)$$

108 The first term in the objective is proportional to the negative log-likelihood, the second is the temporal
 109 penalty for the time-series at each location (i, j) , while the third and fourth, penalize the difference
 110 between the estimated variance of two vertically and horizontally adjacent points, respectively. The
 111 spatial component of this penalty is a special case of trend filtering on graphs [29] which penalizes
 112 the difference between the estimated values of the signal on the connected nodes. As before, we can
 113 write (4) in matrix form where h is an $T \times S$ vector and D is replaced by $D_{TS} \in \mathbb{R}^{(N_t+N_s) \times (T \cdot S)}$,
 114 where $N_t = S \cdot (T - 2)$ and $N_s = T \cdot (2n_r n_c - n_r)$ are the number of temporal and spatial
 115 constraints, respectively¹. Then, as we have two different tuning parameters for the temporal and
 116 spatial components, we write $\Lambda = [\lambda_1 \mathbf{1}_{N_t}^\top, \lambda_2 \mathbf{1}_{N_s}^\top]^\top$ leading to ²:

$$\min_h -y | h) + \Lambda^\top | D_{TS} h | \quad (5)$$

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

118 3 Proposed optimization methods

119 For a spatial grid of size S and T time steps, D_{ST} will have $3Tn_r n_c - 2n_r n_c - Tn_r$ rows and ST
 120 columns. For a $1^\circ \times 1^\circ$ grid over the entire northern hemisphere and daily data over 10 years, we
 121 have $n_r = 90$, $n_c = 180$, $T = 3650$ and so D_{ST} has approximately 10^8 columns and 10^8 rows.
 122 In each step of Algorithm 1, we need to solve a linear system of equations in A which depends on
 123 $D_{ST}^\top D_{ST}$ (see [3] equation 11.54). Therefore, applying the PDIP directly is infeasible for our data.³

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

128 3.1 Consensus optimization

129 Consider an optimization problem of the form $\min_h f(h)$, where $h \in \mathbb{R}^n$ is the *global variable* and
 130 $f(h) : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is convex. The goal in consensus optimization is to break this problem into
 131 several smaller sub-problems which can be solved independently in each iteration of optimization.

132 Assume that it is possible to define a set of *local variables* $x_i \in \mathbb{R}^{n_i}$ such that $f(h) = \sum_i f_i(x_i)$,
 133 where each x_i is a subset of the global variable h . More specifically, each entry of the local variables
 134 corresponds to an entry of the global variable. Therefore we can define a mapping $\mathcal{G}(i, j)$ from the
 135 local variables indices into the global variable indices: $k = \mathcal{G}(i, j)$ means that the j^{th} entry of x_i
 136 is h_k (or $(x_i)_j = h_k$). For ease of notation, define $\tilde{h}_i \in \mathbb{R}^{n_i}$ as $(\tilde{h}_i)_j = h_{\mathcal{G}(i,j)}$. Then, the original
 137 optimization problem is equivalent to the following problem:

¹ N_s is obtained by counting the number of unique constraints at each location and at all times.

²Throughout the paper, we use $|x|$ for both scalars and vectors. For vectors we use this to denote a vector obtained by taking the absolute value of each entry of x .

³We 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.

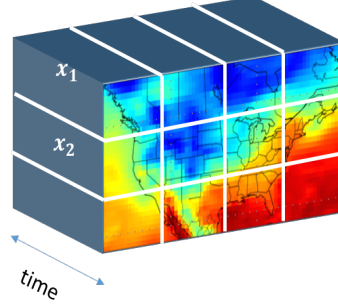


Figure 1: The cube represents the global variable h in space and time. The sub-cubes specified by the white lines are x_i . **TODO: Can we use h_i here instead of x_i ?**

$$\begin{aligned} \min_{\{x_1, \dots, x_N\}} \quad & \sum_i f_i(x_i) \\ \text{s.t.} \quad & \tilde{h}_i = x_i \end{aligned}$$

138 It is important to note that each entry of the global variable may correspond to several entries of
 139 the local variables and so the constraints $\tilde{h}_i = x_i$ enforce the consensus between the local variables
 140 corresponding to the same global variable.

141 The *augmented Lagrangian* corresponding to the problem 3.1 is $L_\rho(x, h, y) = \sum_i (f_i(x_i) + u_i^\top (x_i -$
 142 $\tilde{h}_i) + (\rho/2)\|x_i - \tilde{h}_i\|_2^2)$. Now, we can apply ADMM to L_ρ which results in the following ADMM
 143 updating steps at each iteration m :

$$\begin{aligned} x_i^{m+1} &:= \operatorname{argmin}_{x_i} \left(f_i(x_i) + (u_i^m)^\top x_i + (\rho/2)\|x_i - \tilde{h}_i^m\|_2^2 \right) \\ h_k^{m+1} &:= (1/S_k) \sum_{\mathcal{G}(i,j)=k} (x_i^{m+1})_j \\ u_i^{m+1} &:= u_i^m + \rho(x_i^{m+1} - \tilde{h}_i^{m+1}) \end{aligned} \tag{6}$$

144 Here, S_k is the number of local variable entries that correspond to h_k , and u_i are the Lagrange
 145 multipliers.

146 **TODO: I don't much like this explanation or notation. Seems unclear**

147 **TODO: Arash: I re-wrote this section.**

148 To solve the optimization problem (5) or (4) using this method, we need to address two questions:
 149 first, how to choose the local variables x_i , and second, how to solve the optimization problem for
 150 updating these variables (the first line of (6) which we will refer to it as *x-update step*).

151 In Figure 1, the global variable h is represented as a cube (using the subset of the US as an example).
 152 We decompose h into sub-cubes as shown in the figure by white lines. It is easy to see that by this
 153 definition of x_i , the objective (5) decomposes as $\sum_i f_i(x_i)$ where $f_i(x_i) = -l(y_i | x_i) + \Lambda_{(i)}^\top |D_{(i)} x_i|$,
 154 and $\Lambda_{(i)}$ and $D_{(i)}$ contain the temporal and spatial penalties corresponding to x_i only. The x-update
 155 step in Equation 6 is the following optimization problem: $x_i^{m+1} := \operatorname{argmin}_{x_i} (f_i(x_i) + (u_i^m)^\top x_i +$
 156 $(\rho/2)\|x_i - \tilde{z}_i^m\|_2^2)$.

157 **TODO: I would like to be able to write all this in terms of h instead of x**

158 **TODO: Arash: now I'm using h as the global variables. We cannot write the local variables in terms**
 159 **of h since they are different from h .**

160 By this definition of x_i , the update step for x_i is the following optimization problem: $x_i^{m+1} :=$
 161 $\operatorname{argmin}_{x_i} (f_i(x_i) + (u_i^m)^\top x_i + (\rho/2)\|x_i - \tilde{z}_i^m\|_2^2)$. We solve this using the PDIP method. As it was

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

Input: data y , mapping $\mathcal{G}(i, j)$, ρ , λ_t , λ_s
Initialization: $x_i^0 = z^0 = u_i^0 = \mathbf{0}$.
for $m = 1, 2, \dots$ **do**
 for $i = 1$ **to** $N_{sub-cubes}$ **do**
 compute ν_i from (??)
 compute w_i from (10)
 set $x_i^m := w_i$
 end for
 Compute z^m from (6)
 Compute u_i^m from (6)
end for

162 explained in Section 2.1, to use PDIP we need to compute the dual problem, which in turn needs the
163 computation of the Fenchel conjugate of the loss function. In addition, referring to Algorithm 1, we
164 need to compute the gradient and Jacobian of the conjugate functions. It is important to note that
165 compare to the optimization problem Equation 2, the loss function in the x-update step in Equation 6
166 includes the quadratic term $(\rho/2)\|x_i - \tilde{z}_i^m\|_2^2$. This makes the computation more involved than
167 Section 2.1. The details of the computations are explained in Appendix 1.

168 **TODO: I'm not sure if we want to devote some space to write this as a separate algorithm. The steps**
169 **are those mentioned in 6 and the x-update is now explained in the appendix. So we may want to**
170 **remove algorithm 2.**

171 The complete ADMM algorithm for estimating the variances is represented in Algorithm 2. All the
172 computations in the three updating steps (6) can be performed in parallel. The number of rows and
173 columns of the sub-cubes should be chosen so that the updating of x_i could be performed in one
174 processor. We choose $3 \times 3 \times 521$ sub-cubes.

175 Because Algorithm 2 breaks the large optimization into sub-problems that can be solved independently,
176 it is amenable to a split-gather parallelization strategy via, e.g., the map reduce framework. In each
177 iteration, the computation time will be equal to the time to solve each sub-problem plus the time
178 to communicate the solutions on the master processor and perform the consensus step. Since each
179 sub-problem is small, with parallelization, the computation time in each iteration will be small. In
180 addition, our experiments with several values of λ_t and λ_s showed that the algorithm converges in
181 few hundreds iterations. **TODO: Need to redo this:** Solving each sub-problem on a machine with
182 four 3.20GHz Intel i5-3470 cores takes less than 3 seconds on average, and so for example if we
183 assume that communication time is 10 seconds and the algorithm converges in 300 iterations, with
184 parallelization on $N_{sub-cubes}$ machines, the algorithm will converge in about 1 hour. Assuming that
185 we use $N_{sub-cubes}$ machines and that the convergence rate of the algorithm is independent of the
186 grid size, this time will be independent of the grid size.

187 If we perform these computations on a single machine, the computation time grows linearly with
188 $N_{sub-cubes}$. For example, for the data in a grid over the united states and using $3 \times 3 \times 521$ sub-cubes
189 each iteration of the algorithm will take about 20 minutes on a single machine and so with 300
190 iterations it will take several days to converge. Given that we need to compute the solution for several
191 values of the parameters λ_t and λ_s , this computation time is not feasible.

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

195 3.2 Linearized ADMM

196 **TODO: Need a good dummy variable here, it can't be u .** Consider the generic optimization problem
197 $\min_x f(x) + g(Dx)$ where $x \in \mathbb{R}^n$ and $D \in \mathbb{R}^{m \times n}$. Each iteration of the linearized ADMM

Algorithm 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 $m = 1, 2, \dots$ **do**

$$h_k \leftarrow \mathcal{W} \left(\frac{y_k^2}{\mu} \exp \left(\frac{1 - \mu u_k}{\mu} \right) \right) + \frac{1 - \mu u_k}{\mu},$$

$$z \leftarrow S_{\rho\lambda}(u).$$

$$u \leftarrow u + D h - z$$

end for

198 algorithm [14] for solving this problem has the form

$$\begin{aligned} x &\leftarrow \underset{\mu f}{\text{prox}} \left(x - (\mu/\rho) D^\top (D x - z + u) \right) \\ z &\leftarrow \underset{\rho g}{\text{prox}} (z + u) \\ u &\leftarrow u + D x - z \end{aligned} \tag{7}$$

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

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

200 Clearly, (4) has this form. **TODO: What are μ and ρ ?** To perform the steps in (7), we need to evaluate
201 $\underset{\mu f}{\text{prox}}$ and $\underset{\rho g}{\text{prox}}$. Proximal algorithms are feasible only if these proximal operators can be
202 evaluated efficiently which, as we show next, is the case for our problem.

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

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

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

205 *Proof.* If $f(x) = \sum_k f_k(x_k)$ then $[\underset{\mu f}{\text{prox}}(x)]_k = \underset{\mu f_k}{\text{prox}}(x_k)$. So $[\underset{\mu f}{\text{prox}}(u)]_k =$
206 $\min_{x_k} \mu(x_k + y_k^2 e^{-x_k}) + \frac{1}{2}(x_k - u_k)^2$. Setting the derivative to 0 and solving for u_k gives the
207 result. Similarly, $[\underset{\rho g}{\text{prox}}(u)]_\ell = \rho\lambda_\ell|z_\ell| + 1/2(z_\ell - u_\ell)^2$. This is not differentiable, but the solution
208 must satisfy $\rho \cdot \lambda_\ell \cdot \partial(|z_\ell|) = u_\ell - z_\ell$ where $\partial(|z_\ell|)$ is the sub-differential of $|z_\ell|$. The solution is
209 the soft-thresholding operator $S_{\rho\lambda_\ell}(u_\ell)$. \square

210 4 Empirical evaluation

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

213 4.1 Simulations

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

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

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

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	$\pi/2$



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).

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 $\lambda_t = 100$ (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

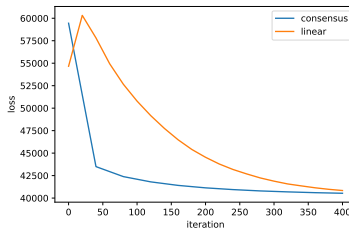


Figure 3: Convergence speed of linearized and consensus ADMM.

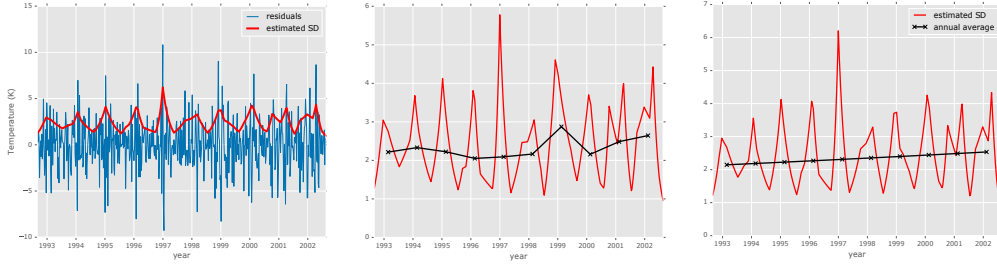


Figure 4: Left: The residuals of the time-series of Bloomington (averaged weekly) and the estimated SD obtained from the method of Section 2.1 (red). Middle: the estimated SDs (red) and their annual average (black) without imposing the long horizon penalty. Right: the same as middle panel but here the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

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):

$$\sum_{i=1}^{N_{year}-2} \left| \sum_{t=1}^{52} h_{t_1} - 2h_{t_2} + h_{t_3} \right| \quad (8)$$

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 (8) adds 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

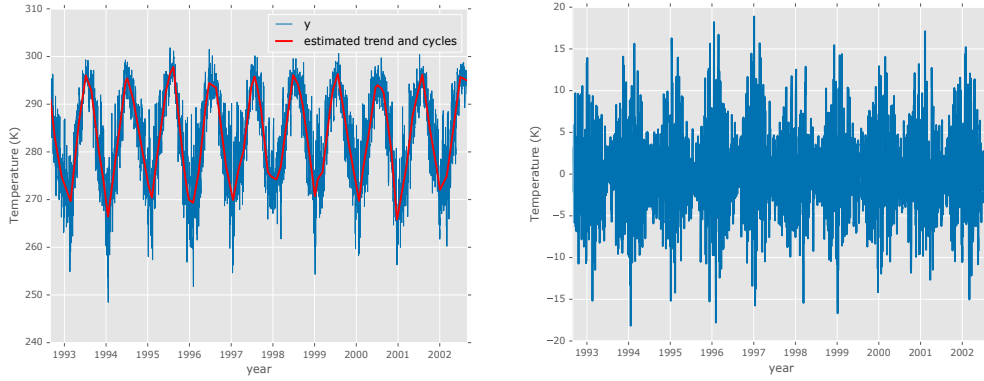


Figure 5: Left: Time-series of the temperature of Bloomington (blue) and the estimated trend and cycles obtained from the ℓ_1 -trend filtering (red). Right: the same time-series after removing the cyclic terms and de-trending using ℓ_1 -trend filtering.

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 $\mathcal{L}(\lambda_t, \lambda_s) = -l(y|h) + \sum \|D_{total}h\|$. 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.

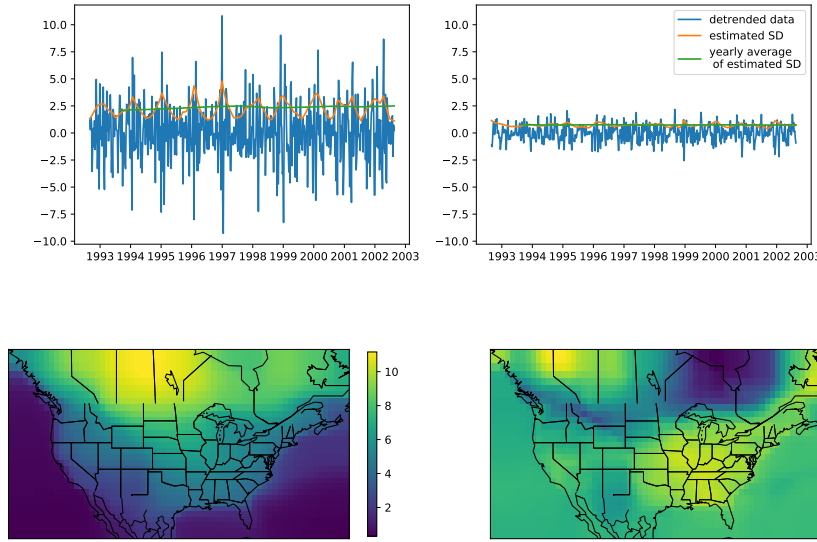


Figure 6: Top row: Detrended data and the estimated SD for Bloomington (left) and San Diego (right). Bottom: the average of the estimated variance over the US (left) and the change in the variance from 1992 to 2002 (right)

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 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.

6 Appendix

TODO: put this in nips appendix format

In this Appendix we give more details on performing the x-update step in Equation 6. We need to solve the following optimization problem:

$$\hat{x} := \operatorname{argmin}_x \left(\sum_{j=1}^{n_b} (x_j + y_j^2 e^{-x_j}) + (\rho/2) \|x - \tilde{z} + u\|_2^2 + \Lambda^\top |Dx| \right)$$

where n_b is the number of local variables in each sub-cube in Figure 1, and for ease of notation we have dropped the subscript i and superscript m . Let $f(x) = \sum_{j=1}^{n_b} (x_j + y_j^2 e^{-x_j}) + (\rho/2) \|x - \tilde{z} + u\|_2^2$. As it was explained in Section 2.1, the dual of this optimization problem is: $\min_{\nu} f^*(-D^\top \nu)$ with the constraints $|\nu_k| \leq \Lambda_k$. So to use PDIP we first need to compute the conjugate function $f^*(\cdot)$. We have:

$$\begin{aligned} f^*(\xi) &= \max_x \xi^\top x - f(x) \\ &= \max_x \sum_{j=1}^{n_b} (\xi_j x_j - x_j - y_j^2 e^{-x_j} - (\rho/2)(x_j - \tilde{z}_j + u_j)) \end{aligned}$$

Setting the derivative of the terms inside the summation to 0, we obtain:

$$\xi_j - y_j^2 e^{-x_j^*} - \rho x_j^* + \rho(\tilde{z}_j - u_j) = 0 \quad (9)$$

where x^* is the maximizer in 6. Then, it can be shown that x_j^* which satisfies (9) can be obtained as follows:

$$\begin{aligned} x_j^* &= \mathcal{W} \left(\frac{y_j^2}{\rho} e^{\phi_j} \right) - \phi_j \\ \phi_j &= \frac{1 - \xi_j - \rho(\tilde{z}_j - u_j)}{\rho} \end{aligned}$$

In this equation, $\mathcal{W}(\cdot)$ is the *Lambert function* [5]. Finally, the conjugate function is: $f^*(\xi) = \sum_{j=1}^{n_b} (\xi_j x_j^* - x_j^* - y_j^2 e^{-x_j^*} - (\rho/2)(x_j^* - \tilde{z}_j + u_j))$.

To use PDIP, we also need to evaluate ∇f^* and $\nabla^2 f^*$. First note that $\frac{\partial \mathcal{W}(q)}{\partial q} = \frac{\mathcal{W}(q)}{q(1+\mathcal{W}(q))}$ and $\frac{\partial^2 \mathcal{W}(q)}{\partial q^2} = -\frac{\mathcal{W}^2(q)(\mathcal{W}(q)+q)}{q^2(1+\mathcal{W}(q))^3}$. Using the chain rule we get:

$$\frac{\partial f^*(\xi)}{\partial \xi_j} = x_j^* + \frac{\partial x_j^*}{\partial \xi_j} \left[\xi_j - 1 + y_j^2 e^{-x_j^*} + \rho(\tilde{z}_j - u_j - x_j^*) \right]$$

where we have:

$$\frac{\partial x_j^*}{\partial \xi_j} = \frac{1}{\rho(1 + \mathcal{W}((y_j^2/\rho)e^{-\phi_j}))}$$

By some tedious but straightforward computation we can obtain the second derivatives:

$$\begin{aligned}\frac{\partial^2 f^*(\xi)}{\partial \xi_j^2} &= \frac{\partial x_j^*}{\partial \xi_j} - \rho \frac{\partial^2 x_j^*}{\partial \xi_j^2} \left[\phi_j + x_j^* - \tilde{z}_j + u_j \right] \\ &\quad + \frac{\partial x_j^*}{\partial \xi_j} \left[1 - y_j^2 \frac{\partial x_j^*}{\partial \xi_j} e^{-x_j^*} - \rho \frac{\partial x_j^*}{\partial \xi_j} \right] \\ \frac{\partial^2 x_j^*}{\partial \xi_j^2} &= \frac{\mathcal{W}((y_j^2/\rho)e^{-\phi_j})}{\rho^2(1 + \mathcal{W}((y_j^2/\rho)e^{-\phi_j}))^3}\end{aligned}$$

Having computed the conjugate function and its gradient and Jacobian, now we can use a number of convex optimization software packages which have an implementation of PDIP to perform the x-update step inside the ADMM loop. We chose the python API of the `cvxopt` [1] package.

References

- [1] M. S. Andersen, J. Dahl, and L. Vandenbergh. 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. Vandenbergh. *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.

- 389 [15] A. Ramdas and R. J. Tibshirani. Fast and flexible admm algorithms for trend filtering. *Journal of*
390 *Computational and Graphical Statistics*, 25(3):839–858, 2016.
- 391 [16] A. Rhines and P. Huybers. Frequent summer temperature extremes reflect changes in the mean, not the
392 variance. *Proceedings of the National Academy of Sciences*, 110(7):E546–E546, Feb. 2013.
- 393 [17] V. Sadhanala, Y.-X. Wang, J. L. Sharpnack, and R. J. Tibshirani. Higher-order total variation classes on
394 grids: Minimax theory and trend filtering methods. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach,
395 R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing*
396 *Systems 30*, pages 5800–5810. Curran Associates, Inc., 2017. URL [http://papers.nips.cc/paper/](http://papers.nips.cc/paper/7162-higher-order-total-variation-classes-on-grids-minimax-theory-and-trend-filtering-methods.pdf)
397 [7162-higher-order-total-variation-classes-on-grids-minimax-theory-and-trend-filtering-methods.](http://papers.nips.cc/paper/7162-higher-order-total-variation-classes-on-grids-minimax-theory-and-trend-filtering-methods.pdf)
398 [pdf](http://papers.nips.cc/paper/7162-higher-order-total-variation-classes-on-grids-minimax-theory-and-trend-filtering-methods.pdf).
- 399 [18] J. A. Screen. Arctic amplification decreases temperature variance in northern mid- to high-latitudes. *Nature*
400 *Climate Change*, 4:577–582, 2014. URL <http://dx.doi.org/10.1038/nclimate2268>.
- 401 [19] P. W. Staten, B. H. Kahn, M. M. Schreier, and A. K. Heidinger. Subpixel characterization of HIRS spectral
402 radiances using cloud properties from AVHRR. *Journal of Atmospheric and Oceanic Technology*, 33(7):
403 1519–1538, 2016. doi: 10.1175/JTECH-D-15-0187.1.
- 404 [20] R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society.*
405 *Series B (Methodological)*, 58(1):267–288, 1996.
- 406 [21] R. J. Tibshirani. *The Solution Path of the Generalized Lasso*. PhD Thesis, Stanford University, 2011.
- 407 [22] R. J. Tibshirani. Adaptive piecewise polynomial estimation via trend filtering. *Annals of Statistics*, 42:
408 285–323, 2014. URL <http://www.stat.cmu.edu/~ryantibs/papers/trendfilter.pdf>.
- 409 [23] R. J. Tibshirani and J. Taylor. The solution path of the generalized lasso. *Annals of Statistics*, 39(3):
410 1335–1371, 2011.
- 411 [24] R. J. Tibshirani and J. Taylor. Degrees of freedom in lasso problems. *The Annals of Statistics*, 40(2):
412 1198–1232, 2012.
- 413 [25] K. E. Trenberth, Y. Zhang, J. T. Fasullo, and S. Taguchi. Climate variability and relationships between
414 top-of-atmosphere radiation and temperatures on earth. *Journal of Geophysical Research: Atmospheres*,
415 120(9):3642–3659, 2014. doi: 10.1002/2014JD022887.
- 416 [26] S. M. Uppala, P. W. K  llberg, A. J. Simmons, U. Andrae, and e. al. The ERA-40 re-analysis. *Quarterly*
417 *Journal of the Royal Meteorological Society*, 131(612):2961–3012, Oct. 2005.
- 418 [27] D. A. Vasseur, J. P. DeLong, B. Gilbert, H. S. Greig, C. D. G. Harley, K. S. McCann, V. Savage, T. D.
419 Tunney, and M. I. O’Connor. Increased temperature variation poses a greater risk to species than climate
420 warming. *Proceedings of the Royal Society of London B: Biological Sciences*, 281(1779), 2014. doi:
421 10.1098/rspb.2013.2612.
- 422 [28] Y.-X. Wang, J. Sharpnack, A. J. Smola, and R. J. Tibshirani. Trend filtering on graphs. *Journal of Machine*
423 *Learning Research*, 17(105):1–41, 2016. URL <http://jmlr.org/papers/v17/15-147.html>.
- 424 [29] Y.-X. Wang, J. Sharpnack, A. J. Smola, and R. J. Tibshirani. Trend Filtering on Graphs. *Journal of Machine*
425 *Learning Research*, 17(105):1–41, 2016. URL <http://jmlr.org/papers/v17/15-147.html>.
- 426 [30] P. Zeng, Q. Hu, and X. Li. Geometry and Degrees of Freedom of Linearly Constrained Generalized Lasso.
427 *Scandinavian Journal of Statistics*, 44(4):989–1008, Nov. 2017. ISSN 0303-6898.