Supplement for "Modeling trend in temperature volatility using generalized LASSO"

Anonymous Author(s)
Affiliation
Address
email

1 Appendix A

In this appendix we provide more details on how to solve the optimization problem 2 using PDIP. The objective function is convex but not differentiable. Therefore, to be able to use PDIP we first need to derive the dual of this problem. We note that this is a generalized LASSO problem [5]. The dual of a generalized LASSO with the objective $f(x) + \lambda \|Dx\|_1$ is:

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

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

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

Each iteration of PDIP involves computing a search direction by taking a Newton step for the system of nonlinear equations $r_w(v, \mu_1, \mu_2) = 0$, where $w > 0$ is a parameter and

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

for $w > 0$, where μ_1 and μ_2 are dual variables for the ℓ_∞ constraint. Let $A = [\nabla r_{dual}^\top, \nabla r_{cent}^\top]^\top$. The newton step takes the following form

$$r_w(v, \mu_1, \mu_2) + A \begin{bmatrix} \nabla v \\ \nabla \mu_1 \\ \nabla \mu_2 \end{bmatrix} = 0$$

We have:

$$A = \begin{bmatrix} \nabla^2 f^*(-D^\top v) & I & -I \\ -\mathbf{diag}(\mu_1)\mathbf{1} & -v + \lambda \mathbf{1} & \mathbf{0} \\ \mathbf{diag}(\mu_2)\mathbf{1} & v + \lambda \mathbf{1} & \mathbf{0} \end{bmatrix}$$

Therefore, to perform the Newton step we need to compute $\nabla f^*(-D^\top v)$ and $\nabla^2 f^*(-D^\top v)$. It is straightforward to show that

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

$$\nabla f^*(-D^\top v) = -\nabla_u f^*(u) D^\top, \quad u = -D^\top v, \quad (\nabla_u f^*(u))_j = \log\left(\frac{y_j^2}{1-u_j}\right)$$
$$\nabla^2 f^*(-D^\top v) = D\nabla_u^2 f^*(u) D^\top, \quad (\nabla_u^2 f^*(u))_j = \mathbf{diag}\left(\frac{1}{1-u_j}\right)$$

2 Appendix B

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 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:

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

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

$$\xi_j - y_j^2 e^{-x_j^*} - \rho x_j^* + \rho(\tilde{z}_j - u_j) = 0 \tag{1}$$

where x^* is the maximizer in 2. Then, it can be shown that x_j^* which satisfies (1) can be obtained as follows:

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

In this equation, $\mathscr{W}(\cdot)$ is the *Lambert function* [4]. 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 $\frac{\partial \mathscr{W}(q)}{\partial q} = \frac{\mathscr{W}(q)}{q(1+\mathscr{W}(q))}$ and $\frac{\partial^2 \mathscr{W}(q)}{\partial q^2} = -\frac{\mathscr{W}^2(q)(\mathscr{W}(q)+q)}{q^2(1+\mathscr{W}(q))^3}$. 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:

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

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	$\pi/2$

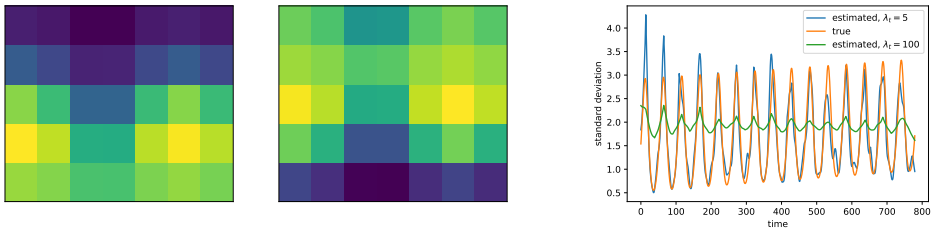

Figure 1: Variance function at $t = 25$ (left) and $t = 45$ (center). Right: the true (orange) and estimated standard deviation function at the location (0,0). The estimated values are obtained using linearized ADMM with $\lambda_s = 0.1$ and two values of λ_t: $\lambda_t = 5$ (blue) and $\lambda_t = 100$ (green).

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

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

$$\frac{\partial^2 x_j^*}{\partial \xi_j^2} = \frac{\mathscr{W}((y_j^2/\rho)e^{-\phi_j})}{\rho^2(1 + \mathscr{W}((y_j^2/\rho)e^{-\phi_j}))^3}$$

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` package [1].

3 Appendix C

Table 1 lists the parameters used for simulating data in Section 4.1. 1 shows the variance function obtained from there parameters at $t = 25$ (left) and $t = 45$ (center).

4 Appendix D

In this appendix we examine some of the properties of the time-series of the temperature in ERA-20C dataset. The goal here is to demonstrate some of the difficulties in modeling the trend in the temperature volatility and motivate our methodology.

2 shows the time-series of the temperature of three cities: Bloomington (USA), San Diego (USA) and Manaus (Brazil). The time-series of Bloomington and San Diego show clear cyclic behavior. However, while it seems (qualitatively) that these cycles can be modeled by a sinusoidal function for Bloomington, the same is not true for San Diego. Also, the amplitude of the cycles changes from some years to others. The time-series of Manaus does not show any regular cyclic behavior. This demonstrates the first difficulty in analyzing the variance of this data: to analyze the variance, we first need to remove the cyclic terms from all time-series. However, there is a lot of variations in the cyclic behavior of the time-series of different locations. In addition, some of these cycles cannot

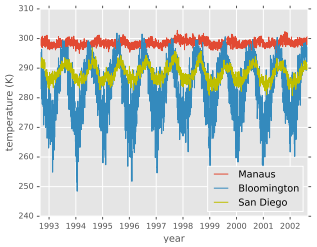

Figure 2: Time-series of the temperature (in Kelvin) of three cities.

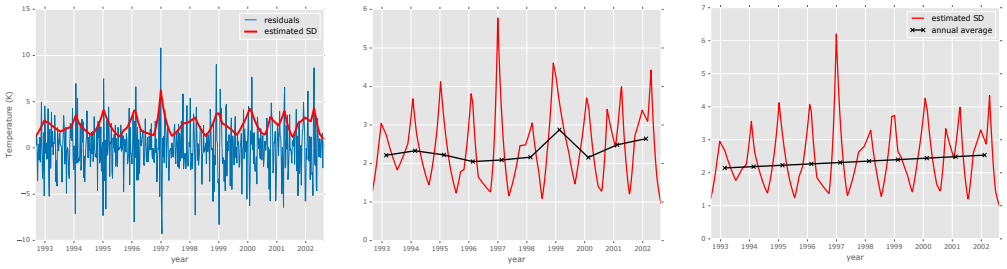

Figure 3: Left: The residuals of the time-series of Bloomington (averaged weekly) and the estimated SD obtained from the method of Section 2.1 (red). Middle: the estimated SDs (red) and their annual average (black) without imposing the long horizon penalty. Right: the same as middle panel but here the long horizon penalty is imposed. See the text for more details.

be easily modeled by a parametric function [1]. To overcome these issues, we use a non-parametric approach to remove the cyclic terms from the time-series and de-trend them. This approach, called ℓ_1*-trend filtering* is explained in Section 2 of the text. We detrended each time-series separately using this method. For each time-series, we found the optimal value of the penalty parameter using *k-fold cross-validation* with $k = 5$. We used the R package **genlasso** to perform these computations [2].

The blue curve in the left panel of 3 shows the time-series of the temperature of Bloomington after detrending using this method. This figure, reveals another difficulty in estimating the trend of volatility in this data: 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 the time-series at all locations using a single parametric model (for example a variant of the GARCH models [3]), 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.

The red curve in the left panel of 3 shows the estimated SD (which is $\exp(h_t/2)$) of the residuals of the time-series of Bloomington obtained from out proposed model. To reduce the number of time-steps 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 3. 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):

[1] One might try to model the cycles by the summation of sinusoidal terms with different frequencies. However, for some time-series this may need many terms to be included in the summation to achieve a reasonable level of accuracy. In addition, this model cannot capture the non-stationarity in the cycles.

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

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 (2) adds N_{year} rows to the matrix D. The estimated SDs using this penalty matrix is shown in the right panel of 3. The annual average of the estimated SDs shows a linear trend with a positive slope.

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. B. Arnold and R. J. Tibshirani. Efficient Implementations of the Generalized Lasso Dual Path Algorithm. *Journal of Computational and Graphical Statistics*, 25(1):1–27, Jan. 2016.

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

[4] 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.

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