Extensions of L1 Trend Filtering: Seasonality

David Johnston

December 31, 2014

Abstract

This paper extends and further elucidates ideas from Kim, Koh & Boyd for L1 regularized trend filtering and acts a documentation to a repository of a python implementation using the cvxopt library which can be found at https://github.com/dave31415/myl1tf

1 The primary and dual quadratic programming problems

We will start off at Section 5.2 of KKB where they state the quadratic programming problem for L1TF and also state the dual problem.

$$\begin{array}{rc} \text{minimize} & \frac{1}{2}\ ||y-x||_2^2 + \lambda ||z||_1 \\ \text{subject to} & z = Dx \end{array} \qquad \begin{array}{r} (1) \\ (2) \end{array}$$

The first is an l_2 norm and the second an l_1 norm. The Lagrangian, with a dual variable $\nu \in \mathbf{R}^{n-2}$, is

$$L(x,z,\nu) = ||y-x||_2^2 + \lambda ||z||_1 + \nu^T(Dx - z)$$

The dual function is

$$\inf_{x,z}\ L(x,z,\nu) = \begin{cases} -\frac{1}{2}\ \nu^T DD^T \nu + y^T D^T \nu & -\lambda\mathbf{1} < \nu < \lambda\mathbf{1} \\ -\infty & \text{otherwise.} \end{cases}$$

and so the dual problem is

$$\begin{array}{rc} \text{maximize} & -\frac{1}{2}\ \nu^T DD^T \nu + y^T D^T \nu \\ \text{subject to} & -\lambda\mathbf{1} < \nu < \lambda\mathbf{1} \end{array} \qquad \begin{array}{r} (3) \\ (4) \end{array}$$

From the solution ν^* of the dual problem, we can compute the L1TF solution,

$$x^* = y - D\nu^*$$

2 Seasonality

As suggested by KKB we can adapt this to add a seasonal component.

$$\begin{aligned}
\text{minimize} \quad & \tfrac{1}{2}\ ||y - x - s||_2^2 + \lambda ||z||_1 & (5)\\
\text{subject to} \quad & z = Dx & (6)\\
\textit{and} \quad & \textstyle\sum_i^P s_i = 0 & (7)\\
\textit{and} \quad & s_{i+P} = s_i & (8)
\end{aligned}$$

KKB does not go into detail on how to proceed with this and so we will begin here by deriving the dual problem. We will do this by putting the constraints on s directly into the equation to be minimized.

To do this, we define p to be the vector of independent variables defining the periodic components. This vector has dimension $(P-1)$ as the P-th, dependent value is $-\sum_{i=1}^{P-1} p_i$ which enforces the constraint that they sum to zero. This constraint is required if there is to exists a unique solution as otherwise one could add any constant to x and subtract it from s without changing the model.

We can define $\tilde{p} \in \mathbf{R}^P$ as $\tilde{p} = \left(p, -\sum_{i=1}^{P-1} p_i\right) = Tp$. T will be a $P \times (P-1)$ matrix with a $(P-1)$ identity matrix at the top and an extra row consisting of all -1. The vector s is now just a periodic re-cycling of $\tilde{p}$ which we can represent as s matrix B which is formed by row-wise stacking some number (ceil(N/P)) of $P \times P$ identity matrices and truncating rows to dimension N. So finally we can write $s = BTp \equiv Qp$. By doing this we can rewrite the optimization problem as

$$\begin{aligned}
\text{minimize} \quad & \tfrac{1}{2}\ ||y - x - Qp||_2^2 + \lambda ||z||_1 & (9)\\
\text{subject to} \quad & z = Dx & (10)
\end{aligned}$$

with the $p \in \mathbf{R}^{(P-1)}$ now unconstrained. To improve stabilization and allow more control over p, we will add a l_2 regularization constraint and write our problem.

$$\begin{aligned}
\text{minimize} \quad & \tfrac{1}{2}\ ||y - x - Qp||_2^2 + \lambda ||z||_1 + \eta\ \tfrac{1}{2}\tilde{p}^T\tilde{p} & (11)\\
\text{subject to} \quad & z = Dx & (12)
\end{aligned}$$

We will now proceed as before and derive the dual problem. To do this we first write down the Lagrangian

$$L(x, z, p, \nu, \eta, \lambda) = R + \lambda ||z||_1 + \nu^T(Dx - z) + \eta\ \frac{1}{2} p^T H\ p$$

with $H = T^T T$ and

$$\begin{aligned}
R \quad &\equiv \quad ||y - x - Qp||_2^2 & (13)\\
&= \quad \frac{1}{2} y^T y + \frac{1}{2} x^T x + \frac{1}{2} p^T Q^T Q p & (14)\\
& \quad\quad - y^T x - y^T Q p + x^T Q p & (15)
\end{aligned}$$

We can calculate $\inf_{x,z,p} L(x, z, p, \nu, \eta, \lambda)$ by setting gradients w.r.t. x and p to zero. The gradients of R are

$$\begin{aligned}
\nabla_x R &= x^T - y^T - p^T Q^T && (16)\\
\nabla_p R &= \left(x^T - y^T - p^T Q^T\right) Q && (17)\\
&= (\nabla_x R)\, Q && (18)\\
& && (19)
\end{aligned}$$

and the gradients of L are

$$\begin{aligned}
\nabla_x L &= x^T - y^T - p^T Q^T + \nu^T D && (20)\\
\nabla_p L &= \left(x^T - y^T - p^T Q^T\right) Q + \nu p^T H && (21)\\
& && (22)
\end{aligned}$$

Setting $\nabla_x L$ =0 yields the equation.

$$y - x - Qp = D^T \nu$$

or

$$x = y - Qp - D^T \nu$$

Equation 2 shows that $D^T \nu$ is the residual.

Setting $\nabla_p L$ =0 yields

$$p = \eta^{-1} H^{-1} Q^T D^T \nu$$

and we can use this last equation for p to solve for x and we can write these solutions as

$$\begin{aligned}
p^* &= \eta^{-1} H^{-1} Q^T D^T \nu && (23)\\
&\mathit{and} &&\\
x^* &= y - D^T \nu - \eta^{-1} Q H^{-1} Q^T D^T \nu && (24)
\end{aligned}$$

and so once we have a solution for ν we can use these to obtain separate solutions for x and p.

The more subtle minimization w.r.t. z will result in the same constraint in the dual problem as before. The reader should ensure that they understand how the terms $\lambda||z||_1 - \nu^T z$ result in the constraint $-\lambda\mathbf{1} < \nu < \lambda\mathbf{1}$. One can show that outside of this range, the inf_z of this term is $-\infty$ (at either $z = \pm\infty$) and within is 0 (at $z = 0$) and so any supremum for ν must lie within.

We then construct the dual problem by plugging these solutions into the Lagrangian and we arrive at

$$\begin{aligned}
\text{maximize} \quad & -\tfrac{1}{2}\, \nu^T A \nu + y^T D^T \nu && (25)\\
\text{subject to} \quad & -\lambda\mathbf{1} < \nu < \lambda\mathbf{1} && (26)
\end{aligned}$$

with

$$A = DD^T + \eta^{-1} DQH^{-1}Q^T D^T$$

We solve this quadratic programming problem as before for ν and then use the equations above to calculate x and p from ν. It is apparent that as $\eta \to \infty$ (seasonality is suppressed), $p \to 0$ and we recover the same solution as before for x. We cannot use these equations directly with the other limit $\eta = 0$, though we will address that in a later section. However there is no problem setting η to a neglibly small number and applying these formula.

3 Seasonality using l_1 regularization

We can also use an l_1 regularization term on the seasonality terms p which will result in sparse solutions for p. That is, with a well chosen regularization parameter, no seasonality will be used when it isn't really required. This might be more useful than the l_2 regularization described above though the solution is a little more complicated.

The situation for l_1 regularization is similar to the above for the x coordinate and leads to the same Equation 2 for the residual. Submitting this solution for x in terms of ν and p leads to an optimization problem for ν and p as follows

$$\begin{aligned} & sup_\nu \ inf_p \quad -\tfrac{1}{2}\ \nu^T DD^T \nu + y^T D^T \nu - \nu^T DQp + \eta ||Tp||_1 && (27) \\ & \text{subject to} \qquad -\lambda \mathbf{1} < \nu < \lambda \mathbf{1} && (28) \end{aligned}$$

We can write $||Tp||_1 = ||p||_1 + |u^T p|$ where u is the unity vector $u_i = 1$. There are no constraints on p so this term $|u^T p|$ can be any non-negative number for some choice of p and so $inf_p \ - \nu^T DQp + \eta ||Tp||_1 = inf_p \ - \nu^T DQp + \eta ||p||_1$ and

$$inf_p \ -\nu^T DQp + \eta ||p||_1 = \begin{cases} 0 & -\eta \mathbf{1} < Q^T D^T \nu < \eta \mathbf{1} \\ -\infty & \text{otherwise.} \end{cases}$$

This implies that the Lagrangian for ν to be optimized is the same as the non-seasonal version but with an additional constraint.

$$\begin{aligned} \text{maximize} \quad & -\tfrac{1}{2}\ \nu^T DD^T \nu + y^T D^T \nu && (29) \\ \text{subject to} \quad & -\lambda \mathbf{1} < \nu < \lambda \mathbf{1} && (30) \\ & -\eta \mathbf{1} < Q^T D^T \nu < \eta \mathbf{1} && (31) \\ & && (32) \end{aligned}$$

Notice that the l_2 solution leads to a modification of the quadratic form whereas the l_1 problem instead leads to an additional constraint. Both of these are standard quadratic programming problems that can be solved with any convex optimization library (such as cvxopt for python). A difference however is that the l_2 solution included a simple way of calculating x and p once ν has been solved. With the l_1 solution here, we have to solve a second optimization problem in order to separate out the separate components.

To do this, note that after ν has been solved for, the first χ^2 term is now constant and so we need to optimize the sum of regularization terms by themselves

$$\begin{aligned} \text{minimize} \quad & ||Dx||_1 + (\lambda/\eta)||T_p||_1 && (33) \\ \text{subject to} \quad & y - x - Qp = D^T\nu && (34) \\ & && (35) \end{aligned}$$

which we can write solely in terms of p as

$$\begin{aligned} \text{minimize over } p: \quad ||D(y - D^T\nu - Qp)||_1 + (\lambda/\eta)||T_p||_1 && (36) \\ && (37) \end{aligned}$$

We can combine these into an extended vector space

$$w = [D(y - D^T\nu), 0]^T$$

$$F = [Qp, -(\lambda/\eta)T]$$

and write this as

$$\begin{aligned} \text{minimize over } p: \quad ||w - Fp||_1 && (38) \\ && (39) \end{aligned}$$

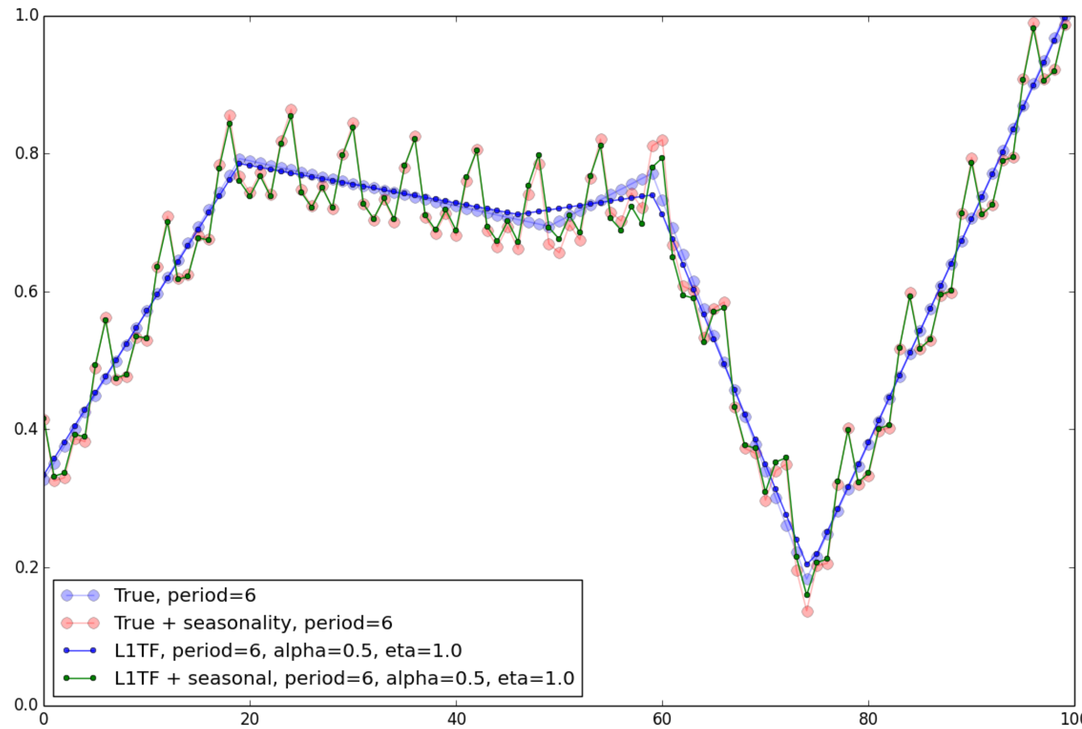
This is now in a standard form of a problem that can be transformed into a linear programming problem. This is the Least Absolute Deviation (LAD) problem. The cvxopt library contains a program for solving this problem. Thus, we can solve for ν by solving the quadratic programming problem and then solve this equation for p and then use 2 to solve for x.

4 Modeling outliers for more robust fits

Finish

5 Implementation

The Githib repository https://github.com/dave31415/myl1tf contains an implementation for this L1TF modeling with seasonality. This is a fork of the repository https://github.com/elsonidoq/py-l1tf by Pablo Zivic which implements the simpler version without seasonality. Both versions are in python and use the python cvxopt library to solve the quadratic programming problems. Our version contains some test programs. For example, the following command,




```
test_myl1tf.test_l1tf_on_mock_with_period(period=6, eta=1.0,alpha=0.5)
```

creates a mock data-set, fits the model and displays the following plot. (Note $eta = \eta$ and $alpha = \lambda$ as *lambda* is a reserved word in python).