Extensions of L1 Trend Filtering: Seasonality, Outlier Modeling and Other Effects with l_1 and l_2 Regularization

David Johnston

March 25, 2015

Abstract

This paper extends and further elucidates ideas from Kim, Koh & Boyd for L1 regularized trend filtering and gives explicit formulas for reducing these problems down to quadratic and linear programming problems that can be solved with various widely available convex optimization libraries. We include seasonality effects with both l_1 and l_2 regularization, outlier modeling, sparse level changes and unequally spaced points. This paper also acts a documentation for the formulas used in our open source python library *myl1tf* [1].

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.

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

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

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

since

[1] https://github.com/dave31415/myl1tf

$$\sup_{\nu} \inf_{z} \left(-\nu^T z + \lambda ||z||_1\right) = \begin{cases} 0 \text{ if } |\nu| < \lambda & \\ -\infty & \text{otherwise.} \end{cases} \tag{3}$$

(see appendix for proof) and so the dual problem is

$$\begin{aligned} \text{maximize} \quad & -\tfrac{1}{2}\ \nu^T D D^T \nu + y^T D^T \nu & (4) \\ \text{subject to} \quad & -\lambda \mathbf{1} < \nu < \lambda \mathbf{1} & (5) \end{aligned}$$

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.

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}\ ||y - x - s||_2^2 + \lambda ||z||_1 & (6) \\ \text{subject to} \quad & z = Dx & (7) \\ \textit{and} \quad & \textstyle\sum_i^P s_i = 0 & (8) \\ \textit{and} \quad & s_{i+P} = s_i & (9) \end{aligned}$$

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

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}\ ||y - x - Qp||_2^2 + \lambda ||z||_1 & (10) \\ \text{subject to} \quad & z = Dx & (11) \end{aligned}$$

with the $p \in \mathbf{R}^{(P-1)}$ now unconstrained.

2.1 Seasonality with l_2 regularization

To improve stabilization and allow more control over p, we will add a l_2 regularization constraint and write our problem.

$$\begin{array}{rc} \text{minimize} & \frac{1}{2}\ ||y - x - Qp||_2^2 + \lambda||z||_1 + \eta\ \frac{1}{2}\tilde{p}^T\tilde{p} \qquad (12) \\ \text{subject to} & z = Dx \qquad (13) \end{array}$$

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

$$\begin{aligned} R &\equiv ||y - x - Qp||_2^2 && (14) \\ &= \frac{1}{2}y^T y + \frac{1}{2}x^T x + \frac{1}{2}p^T Q^T Qp && (15) \\ &\quad -y^T x - y^T Qp + x^T Qp && (16) \end{aligned}$$

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

$$\begin{aligned} \nabla_x R &= x^T - y^T - p^T Q^T && (17) \\ \nabla_p R &= \left(x^T - y^T - p^T Q^T\right) Q && (18) \\ &= (\nabla_x R)\, Q && (19) \\ & && (20) \end{aligned}$$

and the gradients of L are

$$\begin{aligned} \nabla_x L &= x^T - y^T - p^T Q^T + \nu^T D && (21) \\ \nabla_p L &= \left(x^T - y^T - p^T Q^T\right) Q + \nu p^T H && (22) \\ & && (23) \end{aligned}$$

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

$$y - x - Qp = D^T \nu \qquad (24)$$

or

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

Equation 24 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

$$\begin{aligned} p^* &= \eta^{-1} H^{-1} Q^T D^T \nu & (25) \\ & \quad and & \\ x^* &= y - D^T \nu - \eta^{-1} Q H^{-1} Q^T D^T \nu & (26) \end{aligned}$$

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 $z = 0$) (see appendix) and so any supremum for ν must lie within $|\nu| < \lambda$.

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

$$\begin{aligned} \text{maximize} \quad & -\tfrac{1}{2}\ \nu^T A \nu + y^T D^T \nu & (27) \\ \text{subject to} \quad & -\lambda \mathbf{1} < \nu < \lambda \mathbf{1} & (28) \end{aligned}$$

with

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

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.

2.2 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 24 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

$$\begin{aligned} sup_\nu\ inf_p \quad & -\tfrac{1}{2}\ \nu^T DD^T \nu + y^T D^T \nu - \nu^T DQp + \eta ||Tp||_1 & (29) \\ \text{subject to} \quad & -\lambda \mathbf{1} < \nu < \lambda \mathbf{1} & (30) \end{aligned}$$

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.

$$\begin{aligned} \text{maximize} \quad & -\tfrac{1}{2}\ \nu^T DD^T \nu + y^T D^T \nu && (31) \\ \text{subject to} \quad & -\lambda\mathbf{1} < \nu < \lambda\mathbf{1} && (32) \\ & -\eta\mathbf{1} < Q^T D^T \nu < \eta\mathbf{1} && (33) \\ & && (34) \end{aligned}$$

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

$$\begin{aligned} \text{minimize} \quad & ||Dx||_1 + (\eta/\lambda)||Tp||_1 && (35) \\ \text{subject to} \quad & y - x - Qp = D^T \nu && (36) \\ & && (37) \end{aligned}$$

which we can write solely in terms of p as

$$\begin{aligned} \text{minimize over } p: \quad & ||D(y - D^T\nu - Qp)||_1 + (\eta/\lambda)||Tp||_1 && (38) \\ & && (39) \end{aligned}$$

The trick to solving this is to combine these into a larger vector space

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

and the F matrix defined by two matrix blocks

$$F = \left[\frac{DQ}{(\eta/\lambda)T}\right] \qquad (40)$$

and write this as

$$\begin{aligned} \text{minimize over } p: \quad & ||w - Fp||_1 && (41) \\ & && (42) \end{aligned}$$

This is now in the form of a well known problem, the Least Absolute Deviation (LAD) problem and can be written as a a linear program. 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 Equation 24 to solve for x.

3 Modeling outliers and step functions for more robust fits

KKB also suggest including outliers by adding terms u to the model with a strong l_1 regularization. This will result in a sparse solution for the u values which can model things like spikes without throwing off the other terms. These can be treated just like the l_1 regularized seasonality and will result in the same quadratic programming problem as before with an additional constraint $-\delta < D^T\nu < \delta$ where delta is the l_1 regularization parameter for the u vector.

As before, we end up with a solution for ν and must still solve the LAD problem to get separate solutions for x, p and u.

$$\begin{align} \text{minimize over } p, u: \quad & ||D(y - D^T\nu - Qp - u)||_1 + (\lambda/\eta)||Tp||_1 + \delta/\lambda||u||_1 \tag{43} \\ & \tag{44} \end{align}$$

We can employ the same trick as before and write this as a single LAD problem on a larger vector space.

$$\begin{align} \text{minimize over } p, u: \quad & ||w - Fz||_1 \tag{45} \\ & \tag{46} \end{align}$$

with

$$\begin{align} z &= [p, u]^T \tag{47} \\ w &= [D(y - D^T\nu), 0, 0]^T \tag{48} \\ & \tag{49} \end{align}$$

and F specified by matrix blocks

$$F = \left[\begin{array}{c|c} DQ & D \\ \hline (\eta/\lambda)T & 0_{P\times n} \\ \hline 0_{n\times P-1}, & (\delta/\lambda)I_n \end{array}\right] \tag{50}$$

3.1 Step functions

Rather than spikes (δ-functions) we can also consider step functions (or Heaviside functions). The solution is nearly identical to the above except that we replace matrix D in the upper right hand corner with DH where H is the matrix with 1 in it's upper triangle and diagonal and zero in it's lower triangle. This is because step functions are simply cumulative sums of delta functions.

With both spikes u and step function transitions h (regularized by γ), we can write the equations

$$\begin{aligned} z &= [p, u, h]^T && (51) \\ w &= [D(y - D^T \nu), 0, 0, 0]^T && (52) \\ & && (53) \end{aligned}$$

and F specified by matrix blocks

$$F = \left[\begin{array}{c|c|c} DQ & D & DH \\ \hline (\eta/\lambda)T & 0_{P \times n} & 0_{P \times n} \\ \hline 0_{n \times P-1} & (\delta/\lambda)I_n & 0_n \\ \hline 0_{n \times P-1} & 0_n & (\gamma/\lambda)I_n \end{array} \right] \qquad (54)$$

By now, one should notice the pattern that adding any new l_1 regularization terms simply requires a modification of the above equations which just requires that one express the intentions in matrix form.

4 L1 norms for everything

Rather than using an l_2 on the residual we can use an l_1 term for this and so use l_1 norms for everything. An l_1 norm on the residual is another way of giving less weight to outliers. It also simplified the solution as there is no quadratic programming problems to solve. We can use the same trick of writng the whole optimization as a single LAD problem in a higher dimensional space.

Minimize the following for x, p and h:

$$||y - x - Qp - Hh||_1 + \lambda_1 ||D_1 x - g|| + \lambda_2 ||D_2 x|| + \eta ||Tp||_1 + \delta ||h||_1 \qquad (55)$$

Here we are considering first and second derivatives: matrices D_1 and D_2 plus the seaosnality term and level changes. In addition we allow for g a non-zero preferred first derivative. As before we can write this as a single LAD problem:

$$\text{minimize over } z : ||w - Fz||_1$$

where

$$z = [x, p, h]^T \tag{56}$$
$$w = [y, -g, 0, 0, 0]^T \tag{57}$$
$$\tag{58}$$

and F specified by matrix blocks

$$F = \left[\begin{array}{c|c|c} I_n & Q & H \\ \hline -\lambda_1 D_1 & 0_{m\times(P-1)} & 0_{m\times n} \\ \hline -\lambda_2 D_2 & 0_{m\times(P-1)} & 0_{m\times n} \\ \hline 0_{P\times n} & -\eta T & 0_{P\times n} \\ \hline 0_n & 0_{n\times(P-1)} & -\delta I_n \end{array}\right] \tag{59}$$

5 Implementation

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

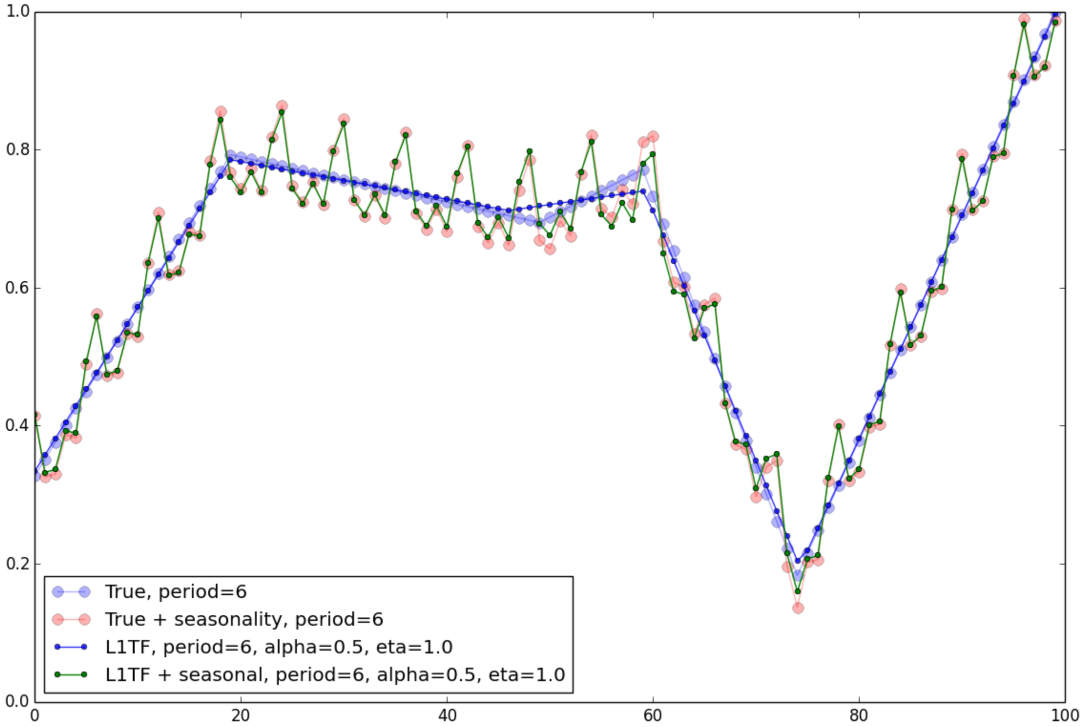
```
test_myl1tf.test_l1tf_on_mock_with_period(period=6, eta=1.0,alpha=0.5)
```

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).

6 Appendix

6.1 Calculating the derivative matrices for unequally spaced points

The first and second derivatives of unequally spaced data points can be calculated, at those data points, as follows. We can use Lagrange's interpolation formula to fit the unique quadratic function through any three points $(x_1, y_1), (x_2, y_2), (x_3, y_3)$.


1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80
100
True, period=6
True + seasonality, period=6
L1TF, period=6, alpha=0.5, eta=1.0
L1TF + seasonal, period=6, alpha=0.5, eta=1.0

$$\begin{aligned} P(x) &= \sum_{i=1}^{3} y_i P_i(x) && (60) \\ P_i(x) &= \prod_{j=1, j\neq i}^{3} \frac{(x-x_j)}{(x_i-x_j)} && (61) \end{aligned}$$

Written out in full this is

$$\begin{aligned} P(x) &= y_1 \left(\frac{x-x_2}{x_1-x_2}\right)\left(\frac{x-x_3}{x_1-x_3}\right) \\ &+ y_2 \left(\frac{x-x_1}{x_2-x_1}\right)\left(\frac{x-x_3}{x_2-x_3}\right) \\ &+ y_3 \left(\frac{x-x_1}{x_3-x_1}\right)\left(\frac{x-x_2}{x_3-x_2}\right) \end{aligned}$$

The first derivative is

$$\begin{aligned} \frac{dP}{dx} &= y_1 \frac{(2x-x_2-x_3)}{(x_1-x_2)(x_1-x_3)} \\ &+ y_2 \frac{(2x-x_1-x_3)}{(x_2-x_1)(x_2-x_3)} \\ &+ y_3 \frac{(2x-x_1-x_2)}{(x_3-x_1)(x_3-x_2)} \end{aligned}$$

and second derivative is

$$\begin{aligned} \frac{d^2P}{dx^2} &= y_1 \frac{2}{(x_1-x_2)(x_1-x_3)} \\ &+ y_2 \frac{2}{(x_2-x_1)(x_2-x_3)} \\ &+ y_3 \frac{2}{(x_3-x_1)(x_3-x_2)} \end{aligned}$$

We are only interested in evaluating these at the middle point $x = x_2$. The second derivative is indpendent of x anyway but the first derivative evalued at $x = x_2$ is

$$\begin{aligned} \frac{dP}{dx}(x=x_2) &= y_1 \frac{(x_2-x_3)}{(x_1-x_2)(x_1-x_3)} \\ &+ y_2 \frac{(2x_2-x_1-x_3)}{(x_2-x_1)(x_2-x_3)} \\ &+ y_3 \frac{(x_2-x_1)}{(x_3-x_1)(x_3-x_2)} \end{aligned}$$

If we assume that the x_i are sorted to be increasing and there are no duplicates, this can be written in matrix form, with first and second derivative matrices F and D being block diagonal with block length 3 and specified by the last two equations.

For equally spaced points of unit separation, i.e. $x_2 = x_1 + 1$ and $x_3 = x_1 + 2$ they reduced to the usual formulas for finite difference derivatives, $F = BlockDiag((-0.5, 0, 0.5))$ and $D = BlockDiag((1, -2, 1))$. These results for unequally spaced points should give results exactly equal to analytical calculations when evaluated on quadratic, linear or constant functions. These two facts provides a useful test for any implementation.

6.2 Proof of Equation 3

We wish to prove that

$$\sup_{\nu} \inf_{z} \left(-\nu^T z + \lambda ||z||_1 \right) = \begin{cases} 0 \text{ if } |\nu| < \lambda & \\ -\infty & \text{otherwise.} \end{cases} \tag{62}$$

Note that this equation is proven if we prove it for each component that is summed so we can drop vector notation and treat them as scalars.

First define $f(\nu) = \inf_z \left(-\nu z + \lambda |z|\right)$. Notice the important fact that $f(\nu) = f(-\nu)$ because if the inf is achieved at z for some ν then the same inf can be achieved at $-z$ for $-\nu$. Thus it is symmetric in ν, $f(\nu) = f(|\nu|)$. Note also that $f(0) = 0$.

Now let's consider the case $|\nu| \leq \lambda$. Clearly $\nu z \leq |\nu z|$ so $f(\nu) = \inf_z \left(-\nu z + \lambda |z|\right) \geq \inf_z \left(|z|(-|\nu| + \lambda)\right) \geq 0$. So $f(\nu) = 0$.

Now consider the other case $|\nu| > \lambda$. Now for any positive z, $f(\nu) < ((-\nu + \lambda)|z|)$ which has $inf_z = -\infty$.

7 References

S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky SIAM Review, problems and techniques section, 51(2):339360, May 2009.
http://stanford.edu/~boyd/papers/pdf/l1_trend_filter.pdf