Sum-of-Norm(SON) Regularization in Estimation Problems

Lennart Ljung
with H. Ohlsson, S. Boyd, F. Gustafsson

Division of Automatic Control
Linköping University
Sweden
Sparse Approximations (Compressed Sensing)

- Given a matrix $A$. Approximate it with a sparse matrix ("many" zero elements) $\hat{A}$.
- Make $\|A - \hat{A}\|_2^2$ small while $\|\hat{A}\|_0$ small ($\|x\|_0 = \#$ of nonzero elements in $x$).
- Various trade-offs controlled by
  $$\min_{\hat{A}} \|A - \hat{A}\|_2^2 + \lambda\|\hat{A}\|_0$$
- Testing $k$ out of $n$ elements to be zero: Difficult combinatorial problem!
Sparse Approximations (Compressed Sensing)

- Given a matrix $A$. Approximate it with a sparse matrix (“many” zero elements) $\hat{A}$.
- Make $\|A - \hat{A}\|_2^2$ small while $\|\hat{A}\|_0$ small ($\|x\|_0 = \#$ of nonzero elements in $x$.)
- Various trade-offs controlled by

\[
\min_{\hat{A}} \|A - \hat{A}\|_2^2 + \lambda \|\hat{A}\|_0
\]

- Testing $k$ out of $n$ elements to be zero: Difficult combinatorial problem!
- Replace the $\ell_0$-norm by the $\ell_1$-norm!

\[
\min_{\hat{A}} \|A - \hat{A}\|_2^2 + \lambda \|\hat{A}\|_1
\]
Linear System with Occasional Disturbances

\[ x(t + 1) = A_t x(t) + B_t u(t) + G_t v(t) \]
\[ y(t) = C_t x(t) + e(t). \]

Here, \( e \) is white measurement noise and \( v \) is a process disturbance.
The Process disturbance, $v$

$v$ is often modeled as Gaussian Noise ... (Kalman filter, LQG etc)
The Process disturbance, $v$

$v$ is often modeled as Gaussian Noise ... (Kalman filter, LQG etc)
But in many applications, $v$ is mostly zero, and strikes only occasionally:

$$v(t) = \delta(t) \eta(t)$$

$$\delta(t) = \begin{cases} 0 & \text{with probability } 1 - \mu \\ 1 & \text{with probability } \mu \end{cases}$$

$$\eta(t) \sim N(0, Q)$$

Examples of applications:

- Control: Load disturbance
- Tracking: Sudden maneuvers
- FDI: Additive system faults
- Recursive Identification ($x=$parameters): model segmentation
Approaches:

- Find the jump times $t$ and/or the smoothed state estimates $\hat{x}_s(t|N)$.

Common methods:
Approaches:

- Find the jump times $t$ and/or the smoothed state estimates $\hat{x}_s(t|N)$.

Common methods:
- Say $\delta(t^*) = 1$ View $t^*$ and $v(t^*)$ as unknown parameters and estimate them. (Willsky-Jones GLR)
Approaches:

- Find the jump times $t$ and/or the smoothed state estimates $\hat{x}_s(t|N)$.

Common methods:

- Say $\delta(t^*) = 1$ View $t^*$ and $v(t^*)$ as unknown parameters and estimate them. (Willsky-Jones GLR)

- Use a CUSUM test to find time instant(s) $t^*$ where the residuals indicate jumps and use Kalman Smoothing with $R_1(t^*) = Q$. 
Approaches:

- Find the jump times $t$ and/or the smoothed state estimates $\hat{x}_s(t|N)$.

Common methods:

- Say $\delta(t^*) = 1$ View $t^*$ and $v(t^*)$ as unknown parameters and estimate them. (Willsky-Jones GLR)

- Use a CUSUM test to find time instant(s) $t^*$ where the residuals indicate jumps and use Kalman Smoothing with $R_1(t^*) = Q$.

- Set $R_1 = \mu Q$ and use Kalman Smoothing to estimate $x$ (and $v(t)$)
Approaches:

- Find the jump times $t$ and/or the smoothed state estimates $\hat{x}_s(t|N)$.

Common methods:
- Say $\delta(t^*) = 1$ View $t^*$ and $v(t^*)$ as unknown parameters and estimate them. (Willsky-Jones GLR)
- Use a CUSUM test to find time instant(s) $t^*$ where the residuals indicate jumps and use Kalman Smoothing with $R_1(t^*) = Q$.
- Set $R_1 = \mu Q$ and use Kalman Smooting to estimate $x$ (and $v(t)$)
- Branch the KF at each time instant: jump/no jump. Prune/merge trajectories (IMM).
Approaches:

- Find the jump times $t$ and/or the smoothed state estimates $\hat{x}_s(t|N)$.

Common methods:

- Say $\delta(t^*) = 1$. View $t^*$ and $v(t^*)$ as unknown parameters and estimate them. (Willsky-Jones GLR)

- Use a CUSUM test to find time instant(s) $t^*$ where the residuals indicate jumps and use Kalman Smoothing with $R_1(t^*) = Q$.

- Set $R_1 = \mu Q$ and use Kalman Smoothing to estimate $x$ (and $v(t)$).

- Branch the KF at each time instant: jump/no jump. Prune/merge trajectories (IMM).

- It is a non-linear filtering problem (linear but not Gaussian noise), so try particle filtering.
Approaches:

- Find the jump times $t$ and/or the smoothed state estimates $\hat{x}_s(t|N)$.

Common methods:

- Say $\delta(t^*) = 1$ View $t^*$ and $v(t^*)$ as unknown parameters and estimate them. (Willsky-Jones GLR)

- Use a CUSUM test to find time instant(s) $t^*$ where the residuals indicate jumps and use Kalman Smoothing with $R_1(t^*) = Q$.

- Set $R_1 = \mu Q$ and use Kalman Smoothing to estimate $x$ (and $v(t)$)

- Branch the KF at each time instant: jump/no jump. Prune/merge trajectories (IMM).

- It is a non-linear filtering problem (linear but not Gaussian noise), so try particle filtering

All methods require some design variables that reflect the trade-off between measurement noise sensitivity and jump alertness.
More om Willsky-Jones GLR

For one jump, estimate $t^*$ and $v(t^*)$ as parameters.

$$x(t + 1) = A_t x(t) + B_t u(t) + G_t v(t)$$
$$y(t) = C_t x(t) + e(t).$$

- If $t^*$ is known it is a simple LS problem to estimate $v(t^*)$. Using the variance of the estimate, the significance of the jump size can be decided in a $\chi^2$ test.
- Find the time of the most significant jump and decide if that is significant enough.
- For detecting several jumps, each detected jump must be accounted for when looking for more.
The Willsky-Jones LS procedure can be written as

Let

$$ W(v(\cdot)) = \sum_{t=1}^{N} \left\| (y(t) - Cx(t)) \right\|^2 $$

such that

$$ x(t+1) = Ax(t) + Bu(t) + Gv(t); \ x(1) = 0. $$

Solve

$$ \min_{v(k), k=1, \ldots, N-1} W(v(\cdot)) $$

s.t. $$ \|V\|_0 = 1; \ V = [\|v(1)\|_2, \ldots, \|v(N-1)\|_2]. $$
The Willsky-Jones LS procedure can be written as

Let

\[ W(v(\cdot)) = \sum_{t=1}^{N} \| (y(t) - Cx(t)) \|^2 \]

such that

\[ x(t + 1) = Ax(t) + Bu(t) + Gv(t); \quad x(1) = 0. \]

Solve

\[ \min_{v(k), k=1, \ldots, N-1} W(v(\cdot)) \]

s.t. \[ \| V \|_0 = 1; \quad V = [\| v(1) \|_2, \ldots, \| v(N - 1) \|_2]. \]

\[ \min_{v(k), k=1, \ldots, N-1} \sum_{t=1}^{N} \| (y(t) - Cx(t)) \|^2 + \lambda \| V \|_0 \]
Do the $\ell_1$ trick!

This problem is computationally forbidding, so relax the $\ell_0$ norm:

$$\min_{v(k), k=1, \ldots, N-1} \sum_{t=1}^{N} \left\| (y(t) - Cx(t)) \right\|^2 + \lambda \left\| V \right\|_1$$

$$= \min_{v(k), k=1, \ldots, N-1} \sum_{t=1}^{N} \left\| (y(t) - Cx(t)) \right\|^2 + \lambda \sum_{t=1}^{N} \left\| v(t) \right\|_2$$

[StateSON]
Do the $\ell_1$ trick!

This problem is computationally forbidding, so relax the $\ell_0$ norm:

$$\min_{v(k), k=1, \ldots, N-1} \sum_{t=1}^{N} \left\| (y(t) - Cx(t)) \right\|^2 + \lambda \|V\|_1$$

$$= \min_{v(k), k=1, \ldots, N-1} \sum_{t=1}^{N} \left\| (y(t) - Cx(t)) \right\|^2 + \lambda \sum_{t=1}^{N} \|v(t)\|_2$$

[StateSON] Compare with Kalman Smoothing:

$$\min_{v(k), k=1, \ldots, N-1} \sum_{t=1}^{N} \left\| R_{e}^{-1/2} (y(t) - Cx(t)) \right\|^2 + \sum_{t=1}^{N} \|R_v^{-1/2}v(t)\|_2$$
Choice of $\lambda$

There is a maximal value of $\lambda$ above which $v(t) \equiv 0$. It can readily be computed as

$$\lambda_{\text{max}} = \max_{k=1,\ldots,N-1} \left\| 2 \sum_{t=k+1}^{N} \left( C A^{t-k-1} G \right)^T \varepsilon_t \right\|_2.$$

where $\varepsilon$ are the no-jump residuals from the system.

Scale by assumed SNR.

Then use $\lambda = \frac{1}{10} \sqrt{\frac{\| R_e \|}{\| Q \|}} \lambda_{\text{max}}$
How does it work?

DC motor with impulse disturbances at $t = 49, 55$. State RMSE over 500 realizations. Dashed blue: Willsky-Jones, Solid green: StateSON
Varying SNRs

Same system. Jump probability $\mu = 0.015$. Varying SNR: $Q =$ jump size, $R_e =$ measurement noise variance. For each SNR, RMSE averages over 500 MC runs. Many different approaches.
Conclusion

- A $\ell_1$ (Sum-of-Norms) relaxation of Willsky-Jones’s estimation problem.
- or The standard ML (Kalman smoother) formulation for smoothing with a quadratic regularization term has been studied for the case without squaring the regularization term
- Still Convex with efficient solution methods
- Favors “sparse” solutions
- Good idea for starting values of the regularization parameter $\lambda$
- Compares favorably with existing solutions
- Many extensions: Model/signal segmentation, path generation, sensor placement, LPV-modeling, Hybrid models.