Consistent dictionary learning for signal declipping
LVA/ICA 2018, Guildford, UK

Lucas Rencker$^1$  Francis Bach$^2$  Wenwu Wang$^1$  Mark D. Plumbley$^1$

$^1$CVSSP, University of Surrey
$^2$SIERRA-team, INRIA

lucas.rencker@surrey.ac.uk

05/07/2018
Table of Contents

1 Background: declipping and dictionary learning

2 Problem formulation

3 Algorithm

4 Experiments

5 Conclusion
Table of Contents

1 Background: declipping and dictionary learning

2 Problem formulation

3 Algorithm

4 Experiments

5 Conclusion
Background: declipping

- Common distortion in signal processing
- Signal saturates above a certain threshold
Background: declipping

Clipping:
- Common distortion in signal processing
- Signal saturates above a certain threshold

Declipping:
- Recovering original signal from clipped signal
- Non-linear, highly under-determined inverse problem (only low energy samples are available)
Background: declipping

Clipping:
- Common distortion in signal processing
- Signal saturates above a certain threshold

Declipping:
- Recovering original signal from clipped signal
- Non-linear, highly under-determined inverse problem (only low energy samples are available)
- Declipping strategies: AR modelling, bandwidth-limited models, Bayesian approaches
More recently: sparsity based approaches:

- \( y \): measured clipped signal
- \( x \): original clean signal

Assume original signal is sparse \( x = D \alpha \), where \( D \in \mathbb{R}^{N \times M} \) (\( N \leq M \))

overcomplete dictionary and \( \| \alpha \|_0 \leq K \).
More recently: **sparsity based approaches**:

- **y**: measured clipped signal
- **x**: original clean signal

- Assume original signal is sparse $x = D \alpha$, where $D \in \mathbb{R}^{N \times M}$ ($N \leq M$) overcomplete dictionary and $\| \alpha \|_0 \leq K$.

- “Straighforward” declipping formulation:

$$\min_{\alpha} \| M^r (y - D \alpha) \|_2^2 \quad \text{s.t.} \quad \| \alpha \|_0 \leq K, \quad (1)$$

where $M^r$ is a binary sensing matrix defining the **reliable** (i.e. unclipped) samples.
More recently: **sparsity based approaches**:

- **\( y \)**: measured clipped signal
- **\( x \)**: original clean signal

Assume original signal is sparse \( x = D \alpha \), where \( D \in \mathbb{R}^{N \times M} \ (N \leq M) \) overcomplete dictionary and \( \| \alpha \|_0 \leq K \).

“Straightforward” declipping formulation:

\[
\min_{\alpha} \| M^r(y - D\alpha) \|_2^2 \quad \text{s.t.} \quad \| \alpha \|_0 \leq K, \quad (1)
\]

where \( M^r \) is a binary sensing matrix defining the reliable (i.e. unclipped) samples.

Many well-known algorithms to solve (1), e.g. (Orthogonal) Matching Pursuit, Iterative Hard Thresholding (IHT), etc...
Example:

- $\hat{\alpha} = \text{argmin}_\alpha \| M^r(y - D\alpha) \|_2^2$ s.t. $\| \alpha \|_0 \leq K$
- estimate full clean signal $\hat{x} = D\hat{\alpha}$:
Example:

- \( \hat{\alpha} = \text{argmin}_\alpha \| M^r(y - D \alpha) \|_2^2 \) s.t. \( \| \alpha \|_0 \leq K \)
- estimate full clean signal \( \hat{x} = D\hat{\alpha} \):
Example:

- $\hat{\alpha} = \arg\min_{\alpha} \| M^r(y - D\alpha) \|_2^2 \quad \text{s.t.} \quad \| \alpha \|_0 \leq K$
- estimate full clean signal $\hat{x} = D\hat{\alpha}$:

- “classical” well known sparse recovery algorithms do not perform well on declipping!
Strategy: enforce reconstructed samples to be above/below the clipping threshold [Adler2012]:

\[
\min_{\alpha} \| M^r(y - D\alpha) \|_2^2 \quad \text{s.t.} \quad \begin{cases} 
\| \alpha \|_0 \leq K \\
M^{c+} D\alpha \geq \theta^+ M^{c+} 1 \\
M^{c-} D\alpha \leq \theta^- M^{c-} 1 
\end{cases}
\]

(2)

where \( M^{c+} \) and \( M^{c-} \) define the position of the positive/negative clipped samples, and \( \theta^+/\theta^- \) positive/negative clipping thresholds.
Strategy: enforce reconstructed samples to be above/below the clipping threshold [Adler2012]:

\[
\min_{\alpha} \| M^r (y - D \alpha) \|_2^2 \quad \text{s.t.} \quad \begin{cases}
\| \alpha \|_0 \leq K \\
M^{c+} D \alpha \geq \theta^+ M^{c+} 1 \\
M^{c-} D \alpha \leq \theta^- M^{c-} 1
\end{cases}
\] (2)

where \( M^{c+} \) and \( M^{c-} \) define the position of the positive/negative clipped samples, and \( \theta^+ / \theta^- \) positive/negative clipping thresholds.

Formulation is consistent with the clipping process (fully models our knowledge about the clipping process).
Strategy: enforce reconstructed samples to be **above/below the clipping threshold** [Adler2012]:

\[
\min_{\alpha} \| M^r(y - D\alpha) \|_2^2 \quad \text{s.t.} \quad \begin{cases} 
\| \alpha \|_0 \leq K \\
M^c+ D\alpha \geq \theta^+ M^{c+} 1 \\
M^c- D\alpha \leq \theta^- M^{c-} 1 
\end{cases}
\]

(2)

where \( M^{c+} \) and \( M^{c-} \) define the position of the positive/negative clipped samples, and \( \theta^+ / \theta^- \) positive/negative clipping thresholds.

- Formulation is **consistent** with the clipping process (fully models our knowledge about the clipping process)

- **Difficult constrained, high-dimensional, non-convex optimization problem!**
ADMM-based sparse declipper: (SPADE) [Kitic, 2015]

$$\min_{\alpha} \| \alpha \|_0 + 1_{C(y)}(D \alpha)$$ (3)

with $1_{C(y)}$ indicator function of the set $C(y)$, and:

$$C(y) \triangleq \{ x \mid M^r y = M^r x, M^{c+} x \succeq M^{c+} y, M^{c-} x \preceq M^{c-} y \}$$ (4)

the constraint set associated with clipped signal $y$. 
ADMM-based sparse declipper: (SPADE) [Kitic,2015]

\[
\min_{\alpha} \| \alpha \|_0 + 1_{C(y)}(D \alpha) \quad (3)
\]

with \(1_{C(y)}\) indicator function of the set \(C(y)\), and:

\[
C(y) \triangleq \{ x | M^r y = M^r x, M^{c+} x \succeq M^{c+} y, M^{c-} x \preceq M^{c-} y \} \quad (4)
\]

the constraint set associated with clipped signal \(y\).

- Alternates between hard-thresholding, and non-orthogonal projection:

\[
\arg\min_{\alpha} \| u - \alpha \|_2^2 + 1_{C(y)}(D \alpha), \quad u \in \mathbb{R}^M \quad (5)
\]

\(\text{Hard to compute when } D \text{ is not a tight frame!} \)\(\text{Heavy computational cost!} \)

\(\text{Unstable (does not converge when sparsity level } K \text{ is fixed)} \)
ADMM-based sparse declipper: (SPADE) [Kitic, 2015]

\[
\min_{\alpha} \| \alpha \|_0 + \mathbb{1}_{C(y)}(D \alpha) \tag{3}
\]

with \( \mathbb{1}_{C(y)} \) indicator function of the set \( C(y) \), and:

\[
C(y) \triangleq \{ x \mid M^r y = M^r x, M^{c+} x \succeq M^{c+} y, M^{c-} x \preceq M^{c-} y \} \tag{4}
\]

the constraint set associated with clipped signal \( y \).

- Alternates between hard-thresholding, and non-orthogonal projection:

\[
\arg\min_{\alpha} \| u - \alpha \|_2^2 + \mathbb{1}_{C(y)}(D \alpha), \quad u \in \mathbb{R}^M \tag{5}
\]

- Hard to compute when \( D \) is not a tight frame! \((D^T D \neq \xi I)\)
ADMM-based sparse declipper: (SPADE) [Kitic, 2015]

$$\min_{\alpha} \|\alpha\|_0 + \mathbb{1}_{C(y)}(D\alpha)$$  \hspace{1cm} (3)

with $\mathbb{1}_{C(y)}$ indicator function of the set $C(y)$, and:

$$C(y) \triangleq \{x | Mr y = Mr x, M^{c+} x \succeq M^{c+} y, M^{c-} x \preceq M^{c-} y\}$$ \hspace{1cm} (4)

the constraint set associated with clipped signal $y$.

- Alternates between hard-thresholding, and non-orthogonal projection:

$$\arg\min_{\alpha} \|u - \alpha\|_2^2 + \mathbb{1}_{C(y)}(D\alpha), \quad u \in \mathbb{R}^M$$ \hspace{1cm} (5)

- Hard to compute when $D$ is not a tight frame! ($D^TD \neq \xi I$)
- Needs to be computed iteratively, using (e.g.) another nested ADMM (Heavy computational cost!)
ADMM-based sparse declipper: (SPADE) [Kitic, 2015]

\[ \min_{\alpha} \| \alpha \|_0 + 1_{C(y)}(D \alpha) \]  

with \( 1_{C(y)} \) indicator function of the set \( C(y) \), and:

\[ C(y) \triangleq \{ x | M^r y = M^r x, M^{c+} x \succeq M^{c+} y, M^{c-} x \preceq M^{c-} y \} \]

the constraint set associated with clipped signal \( y \).

- Alternates between hard-thresholding, and non-orthogonal projection:

\[ \arg\min_{\alpha} \| u - \alpha \|_2^2 + 1_{C(y)}(D \alpha), \quad u \in \mathbb{R}^M \]  

- Hard to compute when \( D \) is not a tight frame! (\( D^T D \neq \xi I \))
- Needs to be computed iteratively, using (e.g.) another nested ADMM (Heavy computational cost!)
- Unstable (does not converge when sparsity level \( K \) is fixed)
Alternative consistent strategies:

- Analysis sparsity models [Kitic, 2015], [Gaultier, 2017]

\[
\min_\alpha \| M r(y - D \alpha) \|_2^2 + \| M c + (\theta + 1 - D \alpha) \|_2^2 + \| M c - (\theta - 1 - D \alpha) \|_2^2 \quad \text{s.t.} \quad \| \alpha \|_0 \leq K, (6)
\]

with \( u^+ = \max(0, u) \) and \( u^- = -u^+ \).

- Quadratic cost when clipping constraint is violated

- Smooth cost function

- Gradient descent based algorithms can be extended ("Consistent IHT for signal declipping" [Kitic, 2013])

- Computationally simple
Alternative consistent strategies:

- Analysis sparsity models [Kitic,2015], [Gaultier,2017]
- $\ell_1$-based constrained formulations [Foucart,2016] ⇒ low performance, still extremely slow

\[
\begin{align*}
\min_{\alpha} & \quad \frac{1}{2} \| M_r (y - D\alpha) \|^2_2 + \frac{1}{2} \| M_c + (\theta + 1 - D\alpha) \|^2_2 + \frac{1}{2} \| M_c - (\theta - 1 - D\alpha) \|^2_2 \\
\text{s.t.} & \quad \| \alpha \|^0_0 \leq K,
\end{align*}
\]

($u^+_t = \max(0, u_t)$ and ($u^-_t = -u^+_t$).
Alternative consistent strategies:

- Analysis sparsity models [Kitic, 2015], [Gaultier, 2017]
- \(\ell_1\)-based constrained formulations [Foucart, 2016] \(\Rightarrow\) low performance, still extremely slow
- Smooth regularizers to enforce clipping consistency [Kitic, 2013], [Siedenburg, 2014]:

\[
\min_{\alpha} \| M^r(y - D\alpha) \|^2_2 + \| M^c^+ (\theta^+ 1 - D\alpha)_+ \|^2_2 \\
+ \| M^c^- (\theta^- 1 - D\alpha)_- \|^2_2 \quad \text{s.t.} \quad \| \alpha \|_0 \leq K,
\]

with \((u)_+ = \max(0, u)\) and \((u)_- = -(u)_+\).
Alternative consistent strategies:

- Analysis sparsity models [Kitic,2015], [Gaultier,2017]
- $\ell_1$-based constrained formulations [Foucart,2016] $\Rightarrow$ low performance, still extremely slow
- Smooth regularizers to enforce clipping consistency [Kitic,2013], [Siedenburg,2014]:

\[
\min_{\alpha} \| M^r (y - D \alpha) \|_2^2 + \| M^c^+ (\theta^+ 1 - D \alpha)_+ \|_2^2 \\
+ \| M^c^- (\theta^- 1 - D \alpha)_- \|_2^2 \\
\text{s.t. } \| \alpha \|_0 \leq K,
\]

with $(u)_+ = \max(0, u)$ and $(u)_- = -(u)_+$.

- Quadratic cost when clipping constraint is violated $(\theta^+ M^c^+ 1 \succeq M^c^+ D \alpha)$
Alternative consistent strategies:

- Analysis sparsity models [Kitic, 2015], [Gaultier, 2017]
- $\ell_1$-based constrained formulations [Foucart, 2016] $\Rightarrow$ low performance, still extremely slow
- Smooth regularizers to enforce clipping consistency [Kitic, 2013], [Siedenburg, 2014]:

$$
\min_{\alpha} \| \mathbf{M}^r (\mathbf{y} - \mathbf{D} \alpha) \|_2^2 + \| \mathbf{M}^c^+ (\theta^+ \mathbf{1} - \mathbf{D} \alpha)_+ \|_2^2 \\
+ \| \mathbf{M}^c^- (\theta^- \mathbf{1} - \mathbf{D} \alpha)_- \|_2^2 \quad \text{s.t.} \quad \| \alpha \|_0 \leq K,
$$

(6)

with $(u)_+ = \max(0, u)$ and $(u)_- = -(u)_+$.

- Quadratic cost when clipping constraint is violated $(\theta^+ \mathbf{M}^c^+ \mathbf{1} \succeq \mathbf{M}^c^+ \mathbf{D} \alpha)$
- **Smooth** cost function
Alternative consistent strategies:

- Analysis sparsity models [Kitic, 2015], [Gaultier, 2017]
- $\ell_1$-based constrained formulations [Foucart, 2016] ⇒ low performance, still extremely slow
- Smooth regularizers to enforce clipping consistency [Kitic, 2013], [Siedenburg, 2014]:

$$
\min_{\alpha} \| M^r(y - D\alpha) \|^2_2 + \| M^c+(\theta^+ 1 - D\alpha)_+ \|^2_2 \\
+ \| M^c-(\theta^- 1 - D\alpha)_- \|^2_2 \quad \text{s.t.} \quad \| \alpha \|_0 \leq K,
$$

(6)

with $(u)_+ = \max(0, u)$ and $(u)_- = -(u)_+$.

- Quadratic cost when clipping constraint is violated
  $(\theta^+ M^c+ 1 \succeq M^c+ D\alpha)$
- Smooth cost function
- Gradient descent based algorithms can be extended ("Consistent IHT for signal declipping" [Kitic, 2013])
Alternative consistent strategies:

- Analysis sparsity models [Kitic, 2015], [Gaultier, 2017]
- $\ell_1$-based constrained formulations [Foucart, 2016] ⇒ low performance, still extremely slow
- Smooth regularizers to enforce clipping consistency [Kitic, 2013], [Siedenburg, 2014]:

\[
\min_{\alpha} \| M^r(y - D\alpha) \|^2_2 + \| M^c(\theta^+ 1 - D\alpha)_+ \|^2_2 \\
+ \| M^c(\theta^- 1 - D\alpha)_- \|^2_2 \quad \text{s.t.} \quad \| \alpha \|_0 \leq K,
\]

with $(u)_+ = \max(0, u)$ and $(u)_- = -(u)_+$.

- Quadratic cost when clipping constraint is violated $(\theta^+ M^c 1 \succeq M^c D\alpha)$
- **Smooth** cost function
- Gradient descent based algorithms can be extended ("Consistent IHT for signal declipping" [Kitic, 2013])
- Computationally simple
Dictionary learning

- All declipping approaches use **fixed dictionaries** (DCT, Gabor)

\[
\min_{D \in \mathcal{D}, \alpha_t} \sum_{t=1}^{T} \|x_t - D\alpha_t\|_2^2 \quad \text{s.t.} \quad \forall t, \|\alpha_t\|_0 \leq K \tag{7}
\]

- Adapt the dictionary to the observed data
- Make use of similarities/correlation between frames \(\{x_t\}_{t=1}^T\)

- Many algorithms to solve (7) (MOD, K-SVD, ... ) in the context of clean/noisy data
- Not addressed in the context of clipped data
Dictionary learning

- All declipping approaches use **fixed dictionaries** (DCT, Gabor)
- Dictionary learning has proved to perform better in many inverse problems (denoising, inpainting, deblurring).

\[
\min_{D \in \mathcal{D}, \alpha} \sum_{t=1}^{T} \| x_t - D \alpha_t \|^2_2 \quad \text{s.t.} \quad \| \alpha_t \|_0 \leq K
\]

- Adapt the dictionary to the observed data
- Make use of similarities/correlation between frames
- Many algorithms to solve (7) (MOD, K-SVD, ...) in the context of clean/noisy data
- Not addressed in the context of clipped data
Dictionary learning

- All declipping approaches use **fixed dictionaries** (DCT, Gabor)
- Dictionary learning has proved to perform better in many inverse problems (denoising, inpainting, deblurring).
- Dictionary learning from clean data \( \{x_t\}_{t=1}^{T} \):

\[
\min_{D \in \mathcal{D}, \alpha_t} \sum_{t=1}^{T} \| x_t - D \alpha_t \|_2^2 \quad \text{s.t.} \quad \forall t, \| \alpha_t \|_0 \leq K
\]
Dictionary learning

- All declipping approaches use **fixed dictionaries** (DCT, Gabor)
- Dictionary learning has proved to perform better in many inverse problems (denoising, inpainting, deblurring).
- Dictionary learning from clean data \( \{x_t\}_{t=1}^T \):

\[
\min_{D \in D, \alpha_t} \sum_{t=1}^{T} \|x_t - D \alpha_t\|_2^2 \quad \text{s.t.} \quad \forall t, \|\alpha_t\|_0 \leq K \tag{7}
\]

- Adapt the dictionary to the observed data
- Make use of similarities/correlation between frames \( \{x_t\}_{t=1}^T \)
Dictionary learning

- All declipping approaches use **fixed dictionaries** (DCT, Gabor)
- Dictionary learning has proved to perform better in many inverse problems (denoising, inpainting, deblurring).
- Dictionary learning from clean data \( \{ x_t \}_{t=1,\ldots,T} \):
  \[
  \min_{D \in \mathcal{D}, \alpha_t} \sum_{t=1}^{T} \| x_t - D \alpha_t \|_2^2 \quad \text{s.t.} \quad \forall t, \| \alpha_t \|_0 \leq K \quad (7)
  \]
  - Adapt the dictionary to the observed data
  - Make use of similarities/correlation between frames \( \{ x_t \}_{t=1,\ldots,T} \)
- Many algorithms to solve (7) (MOD, K-SVD, ...) in the context of clean/noisy data
Dictionary learning

- All declipping approaches use **fixed dictionaries** (DCT, Gabor)
- Dictionary learning has proved to perform better in many inverse problems (denoising, inpainting, deblurring).
- Dictionary learning from clean data $\{x_t\}_{t=1}^{T}$:

$$\min_{D \in \mathcal{D}, \alpha_t} \sum_{t=1}^{T} \| x_t - D \alpha_t \|_2^2 \quad \text{s.t.} \quad \forall t, \| \alpha_t \|_0 \leq K \quad (7)$$

- Adapt the dictionary to the observed data
- Make use of similarities/correlation between frames $\{x_t\}_{t=1}^{T}$
- Many algorithms to solve (7) (MOD, K-SVD, ...) in the context of clean/noisy data
- Not addressed in the context of clipped data
Dictionary learning for declipping?

Dictionary learning often performs many iterations over large datasets, so we need a formulation that is:

- **computationally tractable**
- **stable**
- **does not make any assumption on the dictionary** (tightness etc...)
# Table of Contents

1. Background: declipping and dictionary learning

2. Problem formulation

3. Algorithm

4. Experiments

5. Conclusion
Proposed problem formulation

- Reformulate declipping as a problem of minimizing the distance between the approximated signals $D \alpha_t$, and their feasibility sets $C(y_t)$:

$$\min_{D \in \mathcal{D}, \alpha_t} \sum_t d(D \alpha_t, C(y_t))^2 \quad \text{s.t.} \quad \forall t, \|\alpha_t\|_0 \leq K,$$

with:

$$C(y_t) \triangleq \{x | \mathbf{M}^r y_t = \mathbf{M}^r x, \mathbf{M}^{c+} x \succeq \mathbf{M}^{c+} y_t, \mathbf{M}^{c-} x \preceq \mathbf{M}^{c-} y_t\},$$

and $d(x, C(y))$ is the Euclidean distance between $x$ and the set $C(y)$:

$$d(x, C(y)) = \min_{z \in C(y)} \|x - z\|_2.$$
Proposed problem formulation

- Reformulate declipping as a problem of minimizing the distance between the approximated signals $D\alpha_t$, and their feasibility sets $C(y_t)$:

$$\min_{D \in D, \alpha_t} \sum_t d(D\alpha_t, C(y_t))^2 \quad \text{s.t.} \quad \forall t, \|\alpha_t\|_0 \leq K, \quad (8)$$

with:

$$C(y_t) \triangleq \{x | M^r y_t = M^r x, M^{c+} x \succeq M^{c+} y_t, M^{c-} x \preceq M^{c-} y_t\}, \quad (9)$$

and $d(x, C(y))$ is the **Euclidean distance** between $x$ and the set $C(y)$:

$$d(x, C(y)) = \min_{z \in C(y)} \|x - z\|_2. \quad (10)$$

- Enforces signals to be “close” to their feasibility sets, instead of being exactly in the set.
Proposed problem formulation

- Reformulate declipping as a problem of minimizing the distance between the approximated signals $D\alpha_t$, and their feasibility sets $C(y_t)$:

$$\min_{D \in \mathcal{D}, \alpha_t} \sum_t d(D\alpha_t, C(y_t))^2 \quad \text{s.t.} \quad \forall t, \|\alpha_t\|_0 \leq K,$$

with:

$$C(y_t) \triangleq \{x | M^r y_t = M^r x, M^{c+} x \succeq M^{c+} y_t, M^{c-} x \preceq M^{c-} y_t\},$$

and $d(x, C(y))$ is the Euclidean distance between $x$ and the set $C(y)$:

$$d(x, C(y)) = \min_{z \in C(y)} \|x - z\|_2.$$
Properties of $d(x, C(y))^2$:

$$d(x, C(y))^2 = \min_{z \in C(y)} \|x - z\|_2^2$$ so:

Moreover since $C(y)$ is convex:

$d(x, C(y))^2$ is convex, as a minimum of convex functions over a convex set [Boyd, 2004].
Properties of $d(x, C(y))^2$:

\[
d(x, C(y))^2 = \min_{z \in C(y)} \| x - z \|_2^2
\]

so:

- $d(x, C(y))^2$ is **continuous**
Properties of $d(x, C(y))^2$:

\[
d(x, C(y))^2 = \min_{z \in C(y)} \| x - z \|_2^2
\]
so:
- $d(x, C(y))^2$ is \textit{continuous}

Moreover since $C(y)$ is convex:
- $d(x, C(y))^2$ is \textit{convex}, as a minimum of convex functions over a convex set [Boyd, 2004].
Differentiability of $d(x, C(y))^2 = \min_{z \in C(y)} \|x - z\|_2^2$:

Danskin’s Min-Max theorem [Bonnans, 1998]:

- $C$ a compact set
- $g(x) = \min_{z \in C} \phi(x, z)$
- $\forall z \in \mathbb{R}^N$, $\phi(., z)$ is differentiable with gradient $\nabla_x \phi(x, z)$
- $\phi(x, z)$ and $\nabla_x \phi(x, z)$ are continuous on $\mathbb{R}^N \times \mathbb{R}^N$

If:

- $\arg\min_{z \in C} \phi(x, z) = \{z^*\}$ is unique

Then:

- $g(.)$ is differentiable with gradient:
  \[ \nabla g(x) = \nabla_x \phi(x, z^*). \]  
  \( (11) \)
Differentiability of $d(x, C(y))^2$:

Here:

- $d(x, C(y))^2 = \min_{z \in C(y)} \|x - z\|_2^2$
- $\nabla_x \frac{1}{2} \|x - z\|_2^2 = x - z$
- $\arg\min_{z \in C(y)} \|x - z\|_2^2 \triangleq \Pi_{C(y)}(x)$ orthogonal projection of $x$ onto set $C(y)$.
Differentiability of $d(x, C(y))^2$:

Here:

- $d(x, C(y))^2 = \min_{z \in C(y)} \|x - z\|_2^2$
- $\nabla_x \frac{1}{2} \|x - z\|_2^2 = x - z$
- $\arg\min_{z \in C(y)} \|x - z\|_2^2 \triangleq \Pi_{C(y)}(x)$ orthogonal projection of $x$ onto set $C(y)$.
- $\Rightarrow d(x, C(y))^2$ is differentiable with gradient:

$$\nabla_x \frac{1}{2} d(x, C(y))^2 = x - \Pi_{C(y)}(x) \quad (12)$$
Summary/Comparison with Linear Least Squares:

\[ \mathcal{L}(D \alpha, y) = \frac{1}{2} \| D \alpha - y \|_2^2 \]  

(13)
Summary/Comparison with Linear Least Squares:

\[ \mathcal{L}(D\alpha, y) = \frac{1}{2} \| D\alpha - y \|_2^2 \]  

\[ \text{(13)} \]

▶ Continuous

▶ Continuous

▶ Convex

▶ Differentiable with gradient:

\[ \nabla \alpha \mathcal{L}(D\alpha, y) = D^T(D\alpha - y) \]

▶ Lipschitz gradient

When \( C(y) = \{y\} \) (unclipped signal), the two models are equivalent!

▶ Generalizes the Linear Least Squares cost

▶ Minimizing the proposed cost (14) is as simple as minimizing (13)

▶ Performing consistent sparse declipping is as simple as doing (regular) sparse coding!
Summary/Comparison with Linear Least Squares:

\[ \mathcal{L}(\mathbf{D}, \alpha, \mathbf{y}) = \frac{1}{2} \| \mathbf{D}\alpha - \mathbf{y} \|^2_2 \quad (13) \]

- Continuous
- Convex

When \( \mathcal{C}(\mathbf{y}) = \{\mathbf{y}\} \) (unclipped signal), the two models are equivalent!

Minimizing the proposed cost (14) is as simple as minimizing (13)

Performing consistent sparse declipping is as simple as doing (regular) sparse coding!
Summary/Comparison with Linear Least Squares:

\[ \mathcal{L}(D \alpha, y) = \frac{1}{2} \| D \alpha - y \|_2^2 \] (13)

- Continuous
- Convex
- Differentiable with gradient:
  \[ \nabla \alpha \mathcal{L}(D \alpha, y) = D^T (D \alpha - y) \]
Summary/Comparison with Linear Least Squares:

\[ \mathcal{L}(D \alpha, y) = \frac{1}{2} \| D \alpha - y \|_2^2 \]  

- Continuous
- Convex
- Differentiable with gradient:
  \[ \nabla_{\alpha} \mathcal{L}(D \alpha, y) = D^T(D \alpha - y) \]
- Lipschitz gradient

When \( C(y) = \{y\} \) (unclipped signal), the two models are equivalent!

Minimizing the proposed cost (14) is as simple as minimizing (13)

Performing consistent sparse declipping is as simple as doing (regular) sparse coding!
Summary/Comparison with Linear Least Squares:

\[ \mathcal{L}(D \alpha, y) = \frac{1}{2} \| D \alpha - y \|_2^2 \]  

(13)

- Continuous
- Convex
- Differentiable with gradient:
  \[ \nabla_{\alpha} \mathcal{L}(D \alpha, y) = D^T(D \alpha - y) \]
- Lipschitz gradient
- (Closed-form solution)

When \( C(y) = \{y\} \) (unclipped signal), the two models are equivalent!

Generalizes the Linear Least Squares cost

Minimizing the proposed cost (14) is as simple as minimizing (13)

Performing consistent sparse declipping is as simple as doing (regular) sparse coding!
Summary/Comparison with Linear Least Squares:

\[
\mathcal{L}(D \alpha, y) = \frac{1}{2} \| D \alpha - y \|_2^2 \quad (13)
\]

\[
\mathcal{L}(D \alpha, y) = \frac{1}{2} d(D \alpha, \mathcal{C}(y))^2 \quad (14)
\]

- Continuous
- Convex
- Differentiable with gradient:
  \[ \nabla_{\alpha} \mathcal{L}(D \alpha, y) = D^T (D \alpha - y) \]
- Lipschitz gradient
- (Closed-form solution)

- Continuous
- Convex
- Differentiable with gradient:
  \[ \nabla_{\alpha} \mathcal{L}(D \alpha, y) = D^T (D \alpha - \Pi_{\mathcal{C}(y)}(D \alpha)) \]
- Lipschitz gradient

When \( \mathcal{C}(y) = \{y\} \) (unclipped signal), the two models are equivalent!

Generalizes the Linear Least Squares cost

Minimizing the proposed cost (14) is as simple as minimizing (13)

Performing consistent sparse declipping is as simple as doing (regular) sparse coding!
Summary/Comparison with Linear Least Squares:

\[ \mathcal{L}(D\alpha, y) = \frac{1}{2} \|D\alpha - y\|_2^2 \]  \hspace{1cm} (13)

- Continuous
- Convex
- Differentiable with gradient:
  \[ \nabla_\alpha \mathcal{L}(D\alpha, y) = D^T(D\alpha - y) \]
- Lipschitz gradient
- (Closed-form solution)

\[ \mathcal{L}(D\alpha, y) = \frac{1}{2} d(D\alpha, \mathcal{C}(y))^2 \]  \hspace{1cm} (14)

- Continuous
- Convex
- Differentiable with gradient:
  \[ \nabla_\alpha \mathcal{L}(D\alpha, y) = D^T(D\alpha - \Pi_{\mathcal{C}(y)}(D\alpha)) \]
- Lipschitz gradient

When \( \mathcal{C}(y) = \{y\} \) (unclipped signal), the two models are equivalent!
Summary/Comparison with Linear Least Squares:

\[ \mathcal{L}(D\alpha, y) = \frac{1}{2} \|D\alpha - y\|^2 \]  (13)

\[ \mathcal{L}(D\alpha, y) = \frac{1}{2} d(D\alpha, \mathcal{C}(y))^2 \]  (14)

- Continuous
- Convex
- Differentiable with gradient:
  \[ \nabla_\alpha \mathcal{L}(D\alpha, y) = D^T(D\alpha - y) \]
- Lipschitz gradient
- (Closed-form solution)

- Continuous
- Convex
- Differentiable with gradient:
  \[ \nabla_\alpha \mathcal{L}(D\alpha, y) = D^T(D\alpha - \Pi_{\mathcal{C}(y)}(D\alpha)) \]
- Lipschitz gradient

When \( \mathcal{C}(y) = \{y\} \) (unclipped signal), the two models are equivalent!

- Generalizes the Linear Least Squares cost
Summary/Comparison with Linear Least Squares:

\[
\mathcal{L}(D\alpha, y) = \frac{1}{2} \|D\alpha - y\|^2_2 \quad (13)
\]

- Continuous
- Convex
- Differentiable with gradient:
  \[
  \nabla_\alpha \mathcal{L}(D\alpha, y) = D^T(D\alpha - y)
  \]
- Lipschitz gradient
- (Closed-form solution)

\[
\mathcal{L}(D\alpha, y) = \frac{1}{2} d(D\alpha, C(y))^2 \quad (14)
\]

- Continuous
- Convex
- Differentiable with gradient:
  \[
  \nabla_\alpha \mathcal{L}(D\alpha, y) = D^T(D\alpha - \Pi_{C(y)}(D\alpha))
  \]
- Lipschitz gradient

When \(C(y) = \{y\}\) (unclipped signal), the two models are equivalent!
- Generalizes the Linear Least Squares cost
- Minimizing the proposed cost (14) is as simple as minimizing (13)
Summary/Comparison with Linear Least Squares:

\[ \mathcal{L}(D \alpha, y) = \frac{1}{2} \| D \alpha - y \|_2^2 \]  

(13)

\[ \mathcal{L}(D \alpha, y) = \frac{1}{2} d(D \alpha, C(y))^2 \]  

(14)

- Continuous
- Convex
- Differentiable with gradient:
  \[ \nabla_{\alpha} \mathcal{L}(D \alpha, y) = D^T(D \alpha - y) \]
- Lipschitz gradient
- (Closed-form solution)

- Continuous
- Convex
- Differentiable with gradient:
  \[ \nabla_{\alpha} \mathcal{L}(D \alpha, y) = D^T(D \alpha - \Pi_{C(y)}(D \alpha)) \]
- Lipschitz gradient

When \( C(y) = \{y\} \) (unclipped signal), the two models are equivalent!

- Generalizes the Linear Least Squares cost
- Minimizing the proposed cost (14) is as simple as minimizing (13)
- Performing consistent sparse declipping is as simple as doing (regular) sparse coding!
Table of Contents

1 Background: declipping and dictionary learning

2 Problem formulation

3 Algorithm

4 Experiments

5 Conclusion
Algorithm

$$\min_{D \in \mathcal{D}, \alpha_t} \sum_t d(D \alpha_t, C(y_t))^2 \quad \text{s.t.} \quad \forall t, \| \alpha_t \|_0 \leq K$$
Algorithm

$$\min_{D \in D, \alpha_t} \sum_t d(D \alpha_t, C(y_t))^2 \quad \text{s.t.} \quad \forall t, \|\alpha_t\|_0 \leq K$$

Alternate minimization between sparse coefficients $\alpha_t$ and dictionary $D$:

**Proposed Consistent dictionary learning algorithm:**

Iterate until convergence:

**Sparse coding step:**

for $t = 1, \ldots, T$:

$$\alpha_t \leftarrow \alpha_t + \mu_1 D^T (\Pi_{C(y_t)}(D \alpha_t) - D \alpha_t) \quad \triangleright \text{Gradient descent step}$$

$$\alpha_t \leftarrow \mathcal{H}_K(\alpha_t) \quad \triangleright \text{Hard-thresholding}$$

**Dictionary update step:**

$$D \leftarrow \Pi_D(D + \mu_2 \sum_t (\Pi_{C(y_t)}(D \alpha_t) - D \alpha_t) \alpha_t^T) \quad \triangleright \text{Gradient desc.}$$
Explicit computation of projection operator

- The algorithm requires the computation of projections $\Pi_{C(y)}(D\alpha)$ at each iteration.

- The projection operator $\Pi_{C(y)}(y)$ can be computed in closed form as:

$$\Pi_{C(y)}(y) = Mr_y + Mc + \max(y, D\alpha) + Mc - \min(y, D\alpha).$$

- Simple elementwise maxima (negligible computational cost)

- This also shows that the cost $d(D\alpha, C(y))$ can be written explicitly as:

$$d(D\alpha, C(y))^2 = \|Mr(y - D\alpha)\|^2 + \|Mc + (y - D\alpha)\|^2 + \|Mc - (y - D\alpha)\|^2,$$

- Equivalent to regularization-based methods

- Sparse coding step is equivalent to Consistent IHT [Kitic2013]!
Explicit computation of projection operator

- The algorithm requires the computation of projections $\Pi_{\mathcal{C}(y)}(D\alpha)$ at each iteration.
- The projection operator $\Pi_{\mathcal{C}(y)}(.)$ can be computed in closed form as:

$$\Pi_{\mathcal{C}(y)}(D\alpha) = M^r y + M^{c+} \max(y, D\alpha) + M^{c-} \min(y, D\alpha).$$
Explicit computation of projection operator

- The algorithm requires the computation of projections $\Pi_{C(y)}(D \alpha)$ at each iteration.
- The projection operator $\Pi_{C(y)}(.)$ can be computed in closed form as:

$$\Pi_{C(y)}(D \alpha) = M^r y + M^{c+} \max(y, D \alpha) + M^{c-} \min(y, D \alpha).$$

- Simple elementwise maxima (negligible computational cost)
Explicit computation of projection operator

- The algorithm requires the computation of projections $\Pi_{C(y)}(D\alpha)$ at each iteration.
- The projection operator $\Pi_{C(y)}(.)$ can be computed in closed form as:

$$\Pi_{C(y)}(D\alpha) = M^r y + M^{c+} \max(y, D\alpha) + M^{c-} \min(y, D\alpha).$$

- Simple elementwise maxima (negligible computational cost)
- This also shows that the cost $d(D\alpha, C(y))$ can be written explicitly as:

$$d(D\alpha, C(y))^2 = \| M^r (y - D\alpha) \|_2^2 + \| M^{c+} (y - D\alpha) \|_2^2 + \| M^{c-} (y - D\alpha) \|_2^2,$$
Explicit computation of projection operator

- The algorithm requires the computation of projections $\Pi_{C(y)}(D\alpha)$ at each iteration.
- The projection operator $\Pi_{C(y)}(.)$ can be computed in closed form as:

$$\Pi_{C(y)}(D\alpha) = M^r y + M^{c+} \max(y, D\alpha) + M^{c-} \min(y, D\alpha).$$

- Simple elementwise maxima (negligible computational cost)
- This also shows that the cost $d(D\alpha, C(y))$ can be written explicitly as:

$$d(D\alpha, C(y))^2 = \|M^r(y - D\alpha)\|_2^2 + \|M^{c+}(y - D\alpha)_+\|_2^2 + \|M^{c-}(y - D\alpha)_-\|_2^2,$$

- Equivalent to regularization-based methods
- Sparse coding step is equivalent to Consistent IHT [Kitic2013]!
Table of Contents

1 Background: declipping and dictionary learning

2 Problem formulation

3 Algorithm

4 Experiments

5 Conclusion
Experiments

► Tested on audio signals, $T = 2500$ time frames of size $N = 256$, and dictionaries of size $M = 512$.
► Signal-to-Distortion ratio (SDR), computed on the clipped samples, at different clipping levels

Figure: Comparison with state-of-the-art dictionary learning algorithms
Figure: Comparison with state-of-the-art declipping algorithms
Table of Contents

1 Background: declipping and dictionary learning
2 Problem formulation
3 Algorithm
4 Experiments
5 Conclusion
Conclusion

- Re-formulate the declipping problem as minimizing the distance to a convex feasibility set
- Convex and differentiable cost function, generalizes linear least squares $\Rightarrow$ simple optimization problem.
- Consistent dictionary learning improves compared to consistent sparse coding with fixed dictionary.

$$\mathcal{C}(y) = \{x | Mr y = Mr x, Mc + x \succeq Mc + y, Mc - x \succeq Mc - y\} = \{x | f(x) = y\} = f^{-1}(y)$$

where $f$ is the nonlinear clipping function.

- Extend the proposed method to other nonlinear functions (e.g., quantization, 1-bit sensing)
- Extend other sparse coding/dictionary learning algorithms to optimize the proposed cost function.
Conclusion

- Re-formulate the declipping problem as minimizing the distance to a convex feasibility set.
- Convex and differentiable cost function, generalizes linear least squares $\Rightarrow$ simple optimization problem.
- Consistent dictionary learning improves compared to consistent sparse coding with fixed dictionary.

Future work:

$$C(y) = \{x \mid M^r y = M^r x, M^{c+} x \succeq M^{c+} y, M^{c-} x \preceq M^{c-} y\}$$

$$= \{x \mid f(x) = y\}$$

$$= f^{-1}(y)$$

where $f$ is the nonlinear clipping function.
Conclusion

- Re-formulate the declipping problem as minimizing the distance to a convex feasibility set
- Convex and differentiable cost function, generalizes linear least squares ⇒ simple optimization problem.
- Consistent dictionary learning improves compared to consistent sparse coding with fixed dictionary.

Future work:

\[ C(y) = \{ x \mid M^r y = M^r x, M^{c+} x \succeq M^{c+} y, M^{c-} x \preceq M^{c-} y \} \]

\[ = \{ x \mid f(x) = y \} \]

\[ = f^{-1}(y) \]

where \( f \) is the nonlinear clipping function.

- extend the proposed method to other nonlinear functions (ex: quantization, 1-bit sensing)
Conclusion

- Re-formulate the declipping problem as minimizing the distance to a convex feasibility set
- Convex and differentiable cost function, generalizes linear least squares \( \Rightarrow \) simple optimization problem.
- Consistent dictionary learning improves compared to consistent sparse coding with fixed dictionary.

Future work:

\[
C(y) = \{ x \mid M^r y = M^r x, M^{c+} x \succeq M^{c+} y, M^{c-} x \preceq M^{c-} y \} \\
= \{ x \mid f(x) = y \} \\
= f^{-1}(y)
\]

where \( f \) is the nonlinear clipping function.

- extend the proposed method to other nonlinear functions (ex: quantization, 1-bit sensing)
- extend other sparse coding/dictionary learning algorithms to optimize the proposed cost function
Consistent dictionary learning for signal declipping
L. Rencker, F. Bach, W. Wang, Mark D. Plumbley

Code and audio examples available at:
http://www.cvssp.org/Personal/LucasRencker/software.html
or:
https://github.com/LucasRr
Consistent dictionary learning for signal declipping
L. Rencker, F. Bach, W. Wang, Mark D. Plumbley

Code and audio examples available at:
http://www.cvssp.org/Personal/LucasRencker/software.html
or:
https://github.com/LucasRr

Thank you!