ITERATIVE CAUCHY THRESHOLDING:
REGULARISATION WITH A HEAVY-TAILED PRIOR

Perla Mayo†, Robin Holmes‡ and Alin Achim†

† Visual Information Lab, University of Bristol, Bristol, U.K.
‡ University Hospitals Bristol NHS Foundation Trust, Bristol, U.K.

ABSTRACT
In the machine learning era, sparsity continues to attract significant interest due to the benefits it provides to learning models. Algorithms aiming to optimise the $\ell_0$- and $\ell_1$-norm are the common choices to achieve sparsity. In this work, an alternative algorithm is proposed, which is derived based on the assumption of a Cauchy distribution characterising the coefficients in sparse domains. The Cauchy distribution is known to be able to capture heavy-tails in the data, which are linked to sparse processes. We begin by deriving the Cauchy proximal operator and subsequently propose an algorithm for optimising a cost function which includes a Cauchy penalty term. We have coined our contribution as Iterative Cauchy Thresholding (ICT). Results indicate that sparser solutions can be achieved using ICT in conjunction with a fixed over-complete discrete cosine transform dictionary under a sparse coding methodology.

Index Terms— iterative Cauchy thresholding, ISTA, IHT, sparsity, proximal operator

1. INTRODUCTION
Representation learning (RL) involves the acquisition of features that enable the reconstruction or recovery of a signal of interest. Such features can be learnt in several ways. In dictionary learning it is assumed that an observed signal $y \in \mathbb{R}^M$ can be approximately represented or reconstructed by a linear combination of features in a matrix $A \in \mathbb{R}^{M \times N}$ with a vector of coefficients $x \in \mathbb{R}^N$. This is expressed as:

$$\hat{y} = Ax$$

where $\hat{y} \in \mathbb{R}^M$ is the estimation of the observed signal $y$. The error between this approximation and the true signal is thus given by $L(A, x) = ||y - \hat{y}||_2^2$. In Eq. (1), each coefficient $x_i$ provides the contribution of a column vector (feature) $a_i$ in the reconstruction task for $i = 1, ..., N$. Additionally, along with learning of the vectors in $A$, it is of interest to understand their contribution. The goal is then to recover the coefficients within $x$, which is done through the optimisation of

$$x^* = \arg\min_x L(A, x) = \arg\min_x ||y - Ax||_2^2$$

When the matrix $A$ is over-complete ($N >> M$), the problem in Eq. (2) becomes ill-posed with an infinite number of solutions. To alleviate this, a regularisation term is added in Eq. (2) to enforce sparsity among the elements of the coefficients $x_i$:

$$x^* = \arg\min_x L(A, x) + \lambda \varphi(x)$$

where $\lambda$ is a trade-off parameter between data fidelity and the penalty term.

In Eq. (3), the $\ell_0$-norm is the desired choice for $\varphi(\cdot)$. This norm is ideal since it effectively counts the number of non-zero entries in $x$, therefore, the problem is to find the optimal set of coefficients $x_i$ that will allow the reconstruction of $y$ by using as few as possible column vectors from $A$. Since this is a combinatorial (NP-hard) problem, its optimisation becomes challenging. An alternative to this is to replace the $\ell_0$- by its relaxed version, the $\ell_1$-norm. Furthermore, different penalty functions can be

---

†This work was supported in part by a CONACyT PhD studentship under grant 461322 (to Mayo), in part by the Engineering and Physical Sciences Research Council (EPSRC) under grant EP/R009260/1 (AssenSAR), and in part by a Leverhulme Trust Research Fellowship (to Achim).
obtained if a different approach is followed. When the coefficients are assumed to follow a heavy-tailed distribution centred at 0 it is possible to derive alternative penalties. If a Laplace distribution is assumed for the coefficients, then the penalty function corresponds to the $\ell_1$-norm [1]. The use of heavy tailed distributions has also been considered, e.g. in [2], where it was assumed that coefficients follow the Symmetric $\alpha$-Stable distribution. Our aim is to prove the power of the Cauchy distribution in modelling data with heavy tails, and consequently design novel sparse coding approaches.

The rest of the paper is structured as follows: in section 2 a brief summary of related work is presented along with the motivation behind it, section 3 describes the proposed algorithm in detail, section 4 presents the experiments carried out along with the obtained results. Lastly, a conclusion is presented in section 5 along with a discussion of future work.

2. RELATED WORK

The optimisation of either the $\ell_0$- or $\ell_1$-norms can be done through iterative thresholding approaches such as, e.g., the Iterative Hard Thresholding (IHT) [3] and the Iterative Soft Thresholding (IST) [4] algorithms. They are expressed as shown in equations (4) and (5) respectively.

$$x = \begin{cases} x, & \text{if } |x| > x_0 \\ 0, & \text{if } |x| \leq x_0 \end{cases}$$  \hspace{1cm} (4)

$$x = \begin{cases} x - x_0, & \text{if } x > x_0 \\ x + x_0, & \text{if } x < -x_0 \\ 0, & \text{otherwise} \end{cases}$$  \hspace{1cm} (5)

These approaches achieve sparsity by snapping to zero any value below a given threshold. In the case of IST the values are shrunk by $x_0$. In addition, the function in (5) can be derived by computing the proximal operator of the penalty function, obtained by solving the expression in Eq. (6) below.

$$\text{prox}_{\lambda\varphi}(x) = \arg\min_z (z - x)^2 + \lambda\varphi(z)$$  \hspace{1cm} (6)

For IST, using $\varphi(\cdot) = |\cdot|$ results in the expression in (5) with $x_0 = \frac{\lambda}{2}$. For more details on proximal algorithms, please refer to [5, 6].

2.1. Cauchy distribution as prior for sparsity

As previously said, the penalty function $\varphi(\cdot)$ can be defined from a maximum a posteriori (MAP) approach by assuming the coefficients in $x$ follow a heavy tailed distribution centred at 0. The Laplacian distribution is implicitly chosen when the regularisation term is the $\ell_1$-norm. The Cauchy distribution has been found to be an interesting alternative [7]. It belongs to the family of the $\alpha$-Stable distributions and it is one of the few that has a closed form expression for its probability density function (p.d.f.), which is defined in Eq. (7).

$$p(x) = \frac{\gamma}{\pi(\gamma^2 + (x - \delta)^2)}$$  \hspace{1cm} (7)

The Cauchy distribution is described by two parameters for location ($\delta$) and scale ($\gamma$). The behaviour of these parameters is illustrated in Fig. 2.1. It can be seen that a smaller value for $\gamma$ corresponds to sparser densities and a larger concentration around the location $\delta$. When $\delta = 0$, most values will concentrate around the origin. In addition, the smaller $\gamma$, the more aggressive the sparsity enforcing property of the corresponding algorithm will be.

3. A NEW SHRINKAGE ALGORITHM

The shrinkage algorithm proposed in this work relies on the proximal splitting approach for a Cauchy penalty function. This results in the optimisation of one function at a time. Furthermore, $\varphi(\cdot)$ is solved using the shrinkage operator obtained from the computing of the proximal operator for the Cauchy penalty function. This is further reviewed in section 3.1 whilst the full algorithm is described in detail in section 3.2.
3.1. The Cauchy Proximal Operator

The Cauchy proximal operator is obtained when Eq. (6) is solved using \( \varphi(x) = -\log p(x) \) (from Eq. (7)) and assuming \( \delta = 0 \). Therefore, the proximal operator for this penalty function is computed from:

\[
\text{prox}_{\lambda \varphi}(x) = \arg\min (z - x)^2 + \lambda \varphi(z)
\]

\[
= \arg\min (z - x)^2 - \lambda \log \left( \frac{\gamma}{\pi (\gamma^2 + z^2)} \right)
\]

(8)

up to a constant, which results in:

\[
z^3 - xz^2 + (\gamma^2 + \lambda)z - \gamma^2 x = 0
\]

(9)

Using Cardano’s formula to solve this cubic expression, we obtain:

\[
z = \frac{x}{3} + t
\]

(10)

where

\[
t = \sqrt[3]{-\frac{q}{2} + \sqrt{\Delta}}
\]

\[
\Delta = \frac{q^2}{4} + \frac{p^3}{27}
\]

\[
p = \lambda + \gamma^2 - \frac{x^2}{3}
\]

\[
q = -\frac{2}{27}x^3 + \frac{1}{3}\left(\lambda - 2\gamma^2\right)x
\]

This will provide only one real root when \( \Delta > 0 \), and three real roots otherwise. In the latter case the root with the largest absolute value is chosen.

The shrinkage behaviour offered by the Cauchy proximal operator does not require one to explicitly specify a threshold value, as must be done for IST and IHT. However, the shape of the Cauchy proximal operator still depends on the parameters selection. Such shape is an indicator of how aggressive the cut-off will be. Figure 2(a) shows the impact of the parameter \( \lambda \) in the Cauchy proximal operator, whilst Figure 2(b) does the same for \( \gamma \). It can be seen that \( \lambda \) determines the location of the threshold and \( \gamma \) its shape. A more aggressive cut-off will be exhibited when \( \gamma \to 0 \).

In fact, when \( \gamma \to 0 \), its value will be so small that it will barely contribute anything to the function to minimise. If \( \gamma = 0 \), the function to minimise becomes:

\[
\text{prox}_{\lambda \varphi}(x) = \arg\min (z - x)^2 + \lambda \log(z^2)
\]
Fig. 2. Behaviour of the Cauchy Thresholding algorithm with a fixed value of a) $\gamma = 0.001$, b) $\lambda = 1$.

Fig. 3. Comparison of different thresholding algorithms

which is now a non-smooth non-convex function. Its solutions, however, can be computed following the proximal operator approach once again:

$$z = \begin{cases} 
\frac{x}{2} + \frac{\sqrt{x^2 - 4\lambda}}{2}, & x \geq 2\lambda \\
\frac{x}{2} - \frac{\sqrt{x^2 - 4\lambda}}{2}, & x \leq -2\lambda \\
0, & \text{otherwise}
\end{cases}$$

The shape of ICT under these conditions is shown in figure 2(b): as $\gamma \to 0$, the threshold $\to 2$ ($\lambda = 1$). On the other hand, when $\lambda = 0$, the solutions are given by $\gamma i, -\gamma i, x$, which implies that there is no shrinkage. This comes from the fact that $\lambda = 0$ removes the penalty on the coefficients from $x$, and so the learning relies purely on the reconstruction error.

3.2. Iterative Cauchy Thresholding

Once the proximal operator has been obtained, it is our goal to solve Eq. (3) for the coefficients using the Cauchy penalty function. Explicitly, the cost function is now:

$$f(x) = ||y - Ax||^2_2 - \sum_i \log \left( \frac{\gamma}{\pi(\gamma^2 + x_i^2)} \right)$$

This is solved by means of a proximal splitting approach. Under this framework, the function $f(x) = L(x) + \lambda \phi(x)$ can be optimised by separating $L(x)$ and $\lambda \phi(x)$, therefore, the solutions for each function are projected into the next one. For this, only the gradient of $L(x)$ is computed to solve $L(x)$. Such solution is then passed to the solver for the Cauchy penalty function, which is the shrinkage function given by Eq. (10). The shrinkage is applied in an entry-wise fashion. The process is iterated.
### Table 1. Best PSNR values for reconstruction using the different thresholding algorithms after 200 iterations

| Dataset       | Metric | IHT | IST | ICT |
|---------------|--------|-----|-----|-----|
| MNIST         | PSNR   | 50.70 | 50.70 | 50.70 |
|               | % Non-zeros | 85.07 | 85.06 | 83.83 |
| AT&T Faces    | PSNR   | 47.39 | 47.38 | 47.31 |
|               | % Non-zeros | 98.14 | 99.99 | 75.6  |
| Lena          | PSNR   | 47.26 | 47.26 | 47.26 |
|               | % Non-zeros | 99.80 | 99.98 | 95.18 |
| Shepp-Logan   | PSNR   | 53.77 | 53.76 | 53.77 |
|               | % Non-zeros | 41.99 | 44.74 | 41.12 |

### Table 2. Sparsest results for image reconstruction using the different thresholding algorithms after 200 iterations.

| Dataset       | Metric | IHT | IST | ICT |
|---------------|--------|-----|-----|-----|
| MNIST         | PSNR   | 19.52 | 13.95 | 19.50 |
|               | % Non-zeros | 3.87  | 2.62  | 3.87  |
| AT&T Faces    | PSNR   | 24.02 | 13.59 | 23.99 |
|               | % Non-zeros | 4.12  | 1.66  | 4.11  |
| Lena          | PSNR   | 26.92 | 13.67 | 26.87 |
|               | % Non-zeros | 4.08  | 1.64  | 4.07  |
| Shepp-Logan   | PSNR   | 24.61 | 17.78 | 14.07 |
|               | % Non-zeros | 1.40  | 0.72  | 0.03  |

The performance of ICT has been assessed for a reconstruction task via Sparse Coding (SC). The problem to solve for SC is the same as the one posed in Eq. (3). The performance of ICT is compared against the ones for IST and IHT, where only the penalty function changes during the optimisation accordingly. The dictionary used is the fixed over-complete discrete cosine transform (DCT) using a patch size of 8x8 and a total of 144 atoms. Due to the nature of SC, the image to reconstruct is thus split in $T$ overlapping patches of the same size as the atoms (8x8). There is one vector of sparse coefficients $x$ for every one of these patches. The reconstructed image corresponds to the average of the reconstructed overlapping patches. Further details on sparse coding are beyond the scope of this work and can be found elsewhere [8, 9].

The images to reconstruct correspond to the AT&T faces dataset, the MNIST dataset as well as the Lena and the Shepp-Logan phantom images. The thresholding algorithms were evaluated with varying values for their parameters. To obtain different levels of sparsity the parameter $\lambda$ was modified for the three algorithms. The learning rate $\eta$ was fixed for the three algorithms to 0.005. We randomly chose 500 observations for the MNIST dataset and 50 for the AT&T faces. Tables 1 and 2 report the average results on this reconstruction task for the three thresholding algorithms, PSNR values along with the % of non-zero elements ($\ell_0$-norm) are reported, whilst Figure 4 shows the performance as function of the iterations for the four datasets.

---

1. www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html
2. http://yann.lecun.com/exdb/mnist/
Fig. 4. Average of % of non-zero elements in the coefficients vs PSNR for the three thresholding algorithms for the different datasets: (a) AT&T Faces, (b) MNIST, (C) Lena, and (4) Shepp-Logan.

The performance of ICT is very similar to the one produced by IHT, both achieving better PSNRs than the reconstruction results obtained with IST by a considerable margin, for most levels of sparsity considered. The best PSNR values obtained by the three algorithms correspond to the lowest degree of sparsity for each and vice-versa. It is the latter case the one of interest. The similarity in behaviour between IHT and ICT could be attributed to the shape of their functions. In Fig. 3 it can be seen that the ICT shrinkage function lies between those of both IHT and IST and it affects all coefficients regardless of their amplitude. It is important to note as well that the reconstruction itself corresponds to the average of all estimations obtained for every overlapping patch extracted from the original observations. Hence there will always be a subtle blurring effect affecting the final image. This can be alleviated by employing a different framework, such as convolutional sparse coding [1, 10].

5. CONCLUSIONS AND FUTURE WORK

In this paper we introduced a new algorithm to recover sparse coefficients in regularised inverse problems. The proposed approach is based on the assumption that sparse coefficients can be accurately modelled by Cauchy distributions. Consequently, we introduce the Cauchy proximal operator and describe a corresponding proximal splitting approach, which we prefer to refer to as ICT. As an illustrative example, we chose to address the problem of image reconstruction from learnt sparse representations, and we achieve a performance which compares very favourably with the IHT algorithm. We conjecture that the proposed ICT algorithm has the potential to perform better in other inverse problems, including, e.g. image denoising or deconvolution, due to the optimal penalty applied to large coefficients. This is the focus of our current research endeavours.

6. REFERENCES

[1] Roger Grosse, Rajat Raina, Helen Kwong, and Andrew Y Ng, “Shift-invariance sparse coding for audio classification,” arXiv preprint arXiv:1206.5241, 2012.

[2] Pedram Pad, Farnood Salehi, Elisa Celis, Patrick Thiran, and Michael Unser, “Dictionary learning based on sparse distribution tomography,” in Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017, pp. 2731–2740.
[3] Thomas Blumensath and Mike E Davies, “Iterative thresholding for sparse approximations,” Journal of Fourier analysis and Applications, vol. 14, no. 5-6, pp. 629–654, 2008.

[4] Amir Beck and Marc Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” SIAM journal on imaging sciences, vol. 2, no. 1, pp. 183–202, 2009.

[5] Neal Parikh, Stephen Boyd, et al., “Proximal algorithms,” Foundations and Trends® in Optimization, vol. 1, no. 3, pp. 127–239, 2014.

[6] Patrick L Combettes and Jean-Christophe Pesquet, “Proximal splitting methods in signal processing,” in Fixed-point algorithms for inverse problems in science and engineering, pp. 185–212. Springer, 2011.

[7] T. Wan, N. Canagarajah, and A. Achim, “Segmentation of noisy colour images using Cauchy distribution in the complex wavelet domain,” IET Image Processing, vol. 5, no. 2, pp. 159–170, March 2011.

[8] Bruno A Olshausen and David J Field, “Sparse coding with an overcomplete basis set: A strategy employed by v1?,” Vision research, vol. 37, no. 23, pp. 3311–3325, 1997.

[9] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng, “Efficient sparse coding algorithms,” in Advances in neural information processing systems, 2007, pp. 801–808.

[10] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus, “Deconvolutional networks,” in 2010 IEEE Computer Society Conference on computer vision and pattern recognition. IEEE, 2010, pp. 2528–2535.