Concave losses for robust dictionary learning
Rafael Will M de Araujo, Roberto Hirata, Alain Rakotomamonjy

To cite this version:
Rafael Will M de Araujo, Roberto Hirata, Alain Rakotomamonjy. Concave losses for robust dictionary learning. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, Calgary, Canada. hal-01627533

HAL Id: hal-01627533
https://hal.archives-ouvertes.fr/hal-01627533
Submitted on 1 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
CONCAVE LOSSES FOR ROBUST DICTIONARY LEARNING

Rafael Will M. de Araujo, R. Hirata Jr
University of São Paulo
Institute of Mathematics and Statistics
Rua do Matão, 1010 – 05508-090 – São Paulo-SP, Brazil

Alain Rakotomamonjy
Université de Rouen Normandie
LITIS EA 4108
76800 Saint-Étienne-du-Rouvray, France

ABSTRACT
Traditional dictionary learning methods are based on quadratic convex loss function and thus are sensitive to outliers. In this paper, we propose a generic framework for robust dictionary learning based on concave losses. We provide results on composition of concave functions, notably regarding supergradient computations, that are key for developing generic dictionary learning algorithms applicable to smooth and non-smooth losses. In order to improve identification of outliers, we introduce an initialization heuristic based on undercomplete dictionary learning. Experimental results using synthetic and real data demonstrate that our method is able to better detect outliers, is capable of generating better dictionaries, outperforming state-of-the-art methods such as K-SVD and LC-KSVD.

Index Terms— Robust dictionary learning, outlier detection, concave loss function.

1. INTRODUCTION

Dictionary Learning is an important and widely used tool in Signal Processing and Computer Vision. Its versatility is well acknowledged and it can be applied for denoising or for representation learning prior to classification [1, 2]. The method consists in learning a set of overcomplete elements (or atoms) which are useful for describing examples at hand. In this context, each example is represented as a potentially sparse linear span of the atoms. Formally, given a data matrix composed of \( n \) elements of dimension \( d \), \( X \in \mathbb{R}^{d \times n} \) and each column being an example \( x_i \), the dictionary learning problem is given by:

\[
\min_{D \in \mathbb{R}^{d \times k}, \mathbf{A} \in \mathbb{R}^{k \times n}} \frac{1}{2} \sum_{i=1}^{n} \| x_i - D a_i \|^2 + \Omega_D(D) + \Omega_A(A) (1)
\]

where \( \Omega_D \) and \( \Omega_A \) represent some constraints and/or penalties on the dictionary set \( D \) and the matrix coefficient \( A \), each column being a linear combination coefficients \( a_i \) so that \( x_i \approx D a_i \). Typical regularizers are sparsity-inducing penalty on \( A \), or unit-norm constraint on each dictionary element although a wide variety of penalties can be useful [3, 4, 5].

As depicted by the mathematical formulation of the problem, the learned dictionary \( D \) depends on training examples \( \{x_i\}_{i=1}^{n} \). However, because of the quadratic loss function in the data fitting term, \( D \) is in addition, very sensitive to outlier examples. Our goal here is to address the robustness of the approach to outliers. For this purpose, we consider loss functions that downweight the importance of outliers in \( X \) making the learned dictionary less sensitive to them.

Typical approaches in the literature, that aim at mitigating influence of outliers, use Frobenius norm or component-wise \( \ell_1 \) norm as data-fitting term instead of the squared-Frobenius one [6, 7]. Some works propose loss functions such as the \( \ell_q \) function, with \( q \leq 1 \) function or the capped function \( g(u) = \min(u, \epsilon) \), for \( u > 0 \) [8, 9]. Due to these non-smooth and non-convex loss function, the resulting dictionary learning problem is more difficult to solve than the original one given in Equation (1). As such, authors have developed algorithms based on an iterative reweighted least-square approaches tailored to the loss function \( \ell_q \) or \( \min(u, \epsilon) \) [8, 9].

Our contribution in this paper is: (i) to introduce a generic framework for robust dictionary learning by considering as loss function the composition of the Frobenius norm and some concave loss functions (our framework encompasses previously proposed methods while enlarging the set of applicable loss functions); (ii) to propose a generic majorization-minimization algorithm applicable to concave, smooth or non-smooth loss functions. Furthermore, because the resulting learning problem is non-convex, its solution is sensitive to initial conditions, hence we propose a novel heuristic for dictionary initialization that helps in detecting outliers more efficiently during the learning process.

2. CONCAVE ROBUST DICTIONARY LEARNING

2.1. Framework and Algorithm

In order to robustify the dictionary learning process against outliers, we need a learning problem that puts less empha-
sis on examples that are not “correctly” approximated by the learned dictionary. Hence, we propose the following generic learning problem:

\[
\min_{D,A} \frac{1}{2} \sum_i F(\|x_i - Da_i\|^2_2) + \Omega_D(D) + \Omega_A(A). \tag{2}
\]

where \(F(s)\) is a function over \(\mathbb{R}_{>0}\). Note that in the sequel, we will not focus on the penalty and constraints over the dictionary elements and coefficients \(A\). Hence, we consider them as the classical unit-norm constraint over \(\mathbf{d}_j\) and the \(\ell_1\) sparsity-inducing penalty over \(\{a_i\}\).

Concavity of \(F\) is crucial for robustness as it helps in down-weighting influence of large \(\|x_i - Da_i\|_2\). For instance, if we set \(F(\bullet) = \sqrt{\bullet}\), the above problem is similar to the convex robust dictionary learning proposed by Wang et al. [7]. In order to provide better robustness, our goal is to introduce a non-decreasing concave function over \(\mathbb{R}_{>0}\), such as those used for sparsity-inducing penalties. Typically, \(g(\bullet)\) can be any \(q\)-power, \(q \leq 1\) function \(u^q\), the log function \(\log(\epsilon + u)\), the SCAD function [10], or the capped-\(\ell_1\) function \(\min(u, \epsilon)\), or the MCP function [11]. A key property on \(F\) is that concavity is preserved by the composition of some specific concave functions as proved by the following lemma which proof is omitted for space reasons.

**Lemma 1** Let \(g\) be a non-decreasing concave function on \(\mathbb{R}_{>0}\) and \(h\) be a concave function on a domain \(\Omega\) to \(\mathbb{R}_{>0}\), then \(g \circ h\) is concave. Furthermore, if \(g\) is an increasing function then \(g \circ h\) is strictly concave.

In our framework, \(h\) is the square-root function with \(\Omega = \mathbb{R}_{>0}\). In addition, functions \(g\), such as those given above, are either a concave or strictly concave function and are all non-decreasing, hence \(F = g \circ h\) is concave. Owing to concavity, for any \(u_0\) and \(u\) in \(\mathbb{R}_{>0}\),

\[
F(u) \leq F(u_0) + F'(u_0)(u - u_0)
\]

where \(F'(u_0)\) is an element of the superdifferential of \(F\) at \(u_0\). As \(F\) is concave, the superdifferential is always non-empty and if \(F\) is smooth at \(u_0\), then \(F'(u_0)\) is simply the gradient of \(F\) at \(u_0\). However, since \(F\) is a composition of functions, in a non-smooth case, computing superdifferential is difficult unless the inner function is a linear function [12]. Next lemma provides a key result showing that a supergradient of \(g \circ \sqrt{\bullet}\) can be simply computed using chain rule because \(\sqrt{\bullet}\) is a bijection function on \(\mathbb{R}_{>0}\) to \(\mathbb{R}_{>0}\) and \(g\) is non-decreasing.

**Lemma 2** Let \(g\) a non-decreasing concave function on \(\mathbb{R}_{>0}\) and \(h\) a bijective differentiable concave function on a domain \(\mathbb{R}_{>0}\) to \(\mathbb{R}_{>0}\), then if \(g_1\) is a supergradient of \(g\) at \(z\) then \(g_1 \cdot h'(s)\) is a supergradient of \(g \circ h\) at a point \(s\) so that \(z = h(s)\).

**Algorithm 1** The proposed Robust DL method

**Input:** Data matrix \(X \in \mathbb{R}^{n \times p}\), dictionary size \(k, \lambda, \epsilon, M\).

1. if \((k > d)\) and (use undercomplete initialization) then
2. Initialize \(D\) and \(A\) with Algorithm 2
3. else
4. random initialization of \(D, A\)
5. \(s_j = 1\) for \(j = 1, \ldots, n\)
6. for \(i = 1\) to \(M\) do
7. repeat
8. Update \(D\) by solving Equation 5
9. for \(j = 1\) to \(n\) do
10. \(a_j = \frac{1}{\epsilon} \|x_j - Da_i\|^2 + \frac{1}{\epsilon} |a|_1\)
11. until convergence
12. for \(j = 1\) to \(n\) do
13. update \(s_j\) according to Equation 4

**Output:** \(D, A\)

**Proof** As \(g_1 \in \partial g(z)\), we have \(\forall \gamma, g(y) \leq g(z) + g_1 \cdot (y - z)\). Owing to bijectivity of \(h\), define \(t\) and \(s\) so that \(y = h(t)\) and \(z = h(s)\). In addition, concavity of \(h\) gives \(h(t) - h(s) \leq h'(s)(t - s)\) and because \(g\) is non-decreasing, \(g_1 \geq 0\). Combining everything, we have \(g_1 \cdot (y - z) = g_1 \cdot (h(t) - h(s)) \leq g_1 h'(s)(t - s)\). Thus \(\forall t, g(h(t)) \leq g(h(s)) + g_1 h'(s)(t - s)\) which concludes the proof since \(g_1\) is a supergradient of \(g\) at \(h(s)\).

Based on the above majorizing linear function property of concave functions and because in our case \(F'(u_0)\) can easily be computed, we consider a majorization-minimization approach for solving Problem (2). Our iterative algorithm consists, at iteration \(\kappa\), in approximating the concave loss function \(F\) at the current solution \(D_\kappa\) and \(A_\kappa\) and then solve the resulting approximate problem for \(D\) and \(A\). This yields in solving:

\[
\min_{D,A} \frac{1}{2} \sum_i s_i \|x_i - Da_i\|^2_2 + \Omega_D(D) + \Omega_A(A) \tag{3}
\]

where \(s_i = \parallel g \circ \sqrt{\bullet}\parallel\) at \(D_\kappa\) and \(a_{\kappa,i}\). Since, we have

\[
[g \circ \sqrt{\bullet}](u_0) = \frac{1}{2\sqrt{u_0}} g'(\sqrt{u_0})
\]

weights \(s_i\) can be defined as

\[
s_i = \frac{g'(\|x_i - Da_i\|_2)}{2\|x_i - Da_i\|_2}.
\]

This definition of \(s_i\) can be nicely interpreted. Indeed, if \(g\) is so that \(g'(u)\) becomes small as \(u\) increases, examples with large residual values \(\|x_i - Da_i\|_2\) have less importance in the learning problem (3) because their corresponding values \(s_i\) are small.

Note how the composition \(g \circ \sqrt{\bullet}\) allows us to write the data fitting term with respect to the squared residual norm so that at each iteration, the problem (3) to solve is simply a weighted smooth dictionary learning problem, convex in each
of its parameters, that can be addressed using off-the-shelf tools. As such, it can be solved alternatively for \( \mathbf{D} \) with fixed \( \mathbf{A} \) and then for \( \mathbf{A} \) with fixed \( \mathbf{D} \). For fixed \( \mathbf{A} \), the optimization problem is thus:

\[
\min_{\mathbf{D}} \frac{1}{2} \sum_i \| \mathbf{x}_i - \mathbf{D} \hat{\mathbf{a}}_i \|_2^2 + \Omega_D(\mathbf{D}) \tag{5}
\]

where \( \hat{\mathbf{x}}_i = \sqrt{s_i} \mathbf{x}_i \) and \( \hat{\mathbf{a}}_i = \sqrt{s_i} \mathbf{a}_i \). This problem can be solved using a proximal gradient algorithm or block-coordinate descent algorithm as given in Mairal et al. [2]. For fixed \( \mathbf{D} \), the problem is separable in \( \mathbf{a}_i \) and each sub-problem is equivalent to a Lasso problem with regularization \( \frac{\lambda}{s_i} \).

The above algorithm is generic in the sense that it is applicable to any continuous concave and non-decreasing function \( g \), even non-smooth ones. This is in contrast with algorithms proposed in [8, 9] which have been tailored to some specific functions \( g \). In addition, the convergence in objective value of the algorithm is guaranteed for any of these \( g \) functions, by the fact that the objective value in Equation 2 decreases at each iteration while it is obviously lower bounded.

2.2. Heuristic for initialization

The problem we are solving is a non-convex problem and its solution is thus very sensitive to initialization. The presence of outliers in the data matrix \( \mathbf{X} \) magnifies this sensitivity and increases the need for a proper initialization of \( s_i \) in our iterative algorithm based on Equation (3). If we were able to identify outliers before learning, then we would assign \( s_i = 0 \) to these samples so that they become irrelevant for the dictionary learning problem. However, detecting outliers in a set of samples is a difficult learning problem by itself [13].

Our initialization heuristic is based on the intuition that if most examples belong to a linear subspace of \( \mathbb{R}^d \) while outliers leave outside this subspace, then these outliers can be better identified by using an undercomplete dictionary learning than an overcomplete one. Indeed, if the sparsity penalty is weak enough, then an overcomplete dictionary can approximate well any example leading to a large value of \( s_i \) even the for outliers.

Hence, if the number of dictionary to learn is larger than the dimension of the problem, we propose to initialize \( \mathbf{D} \) and \( \mathbf{s} \) by learning mini-batches of size \( b < k \) of dictionary atoms using one iteration of Alg. 1 initialized with \( s_i = 1, \forall i \in [1, \ldots, n] \), a random dictionary and random weights \( \mathbf{A} \). If we have only a small proportions of outliers, we make the hypothesis that the learning problem will focus on dictionary atoms that span a subspace that better approximates non-outlier samples. Then, as each set of learned mini-batch dictionary atoms leads to a different error \( \| \mathbf{x}_i - \mathbf{D} \mathbf{a}_i \|_2 \) and thus to different \( s_i \) as defined in Equation 3, we estimate \( s_i \) as the average \( \bar{s}_i \) over the number of mini-batch and we expect \( s_i \) to be small if \( i \)-th example is an outlier. This initialization strategy is presented in Alg. 2.

![Figure 1](image-url)

### Algorithm 2 Undercomplete initialization

**Input:** Data matrix \( \mathbf{X} \), dictionary \( \mathbf{D} \in \mathbb{R}^{d \times k} \), with \( d < k \), number of atoms in each batch \( b < k \), parameters \( \lambda \) and \( \epsilon \).

1: \( N \leftarrow \left\lfloor \frac{n}{b} \right\rfloor \) \{number of batches\}
2: \( s = 0 \)
3: Initialize \( \mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_k] \) as a zero matrix
4: for \( i = 0 \) to \((N-1)\) do
5: \( I \leftarrow \text{indices related to } i \)-th batch
6: \( \hat{\mathbf{D}}_i, \hat{\mathbf{s}}_i \leftarrow \text{Algorithm 1} \{\mathbf{X}_i, |I|, \lambda, \epsilon, 1\}
7: \( \mathbf{D}_i, \mathbf{s}_i \leftarrow \hat{\mathbf{D}}_i \{\text{assign learned dictionary to the appropriate indices}\}
8: \( s \leftarrow s + \hat{\mathbf{s}}_i \{\text{accumulate weights}\}
9: s \leftarrow \frac{s}{I} \{\text{compute average}\}

**Output:** \( \hat{\mathbf{D}}, \mathbf{s} \)

3. EXPERIMENTS

3.1. Experiments on synthetic data

We use synthetic generated datasets with outliers to demonstrate that our method is robust against outliers. Figure 1 presents two clusters generated from two Gaussian distributions, each containing 250 points along with 50 outliers represented as the red triangles, far away from the clusters. Figure 1 also shows the clustering results using K-SVD [14] as well as the proposed method when \( g(u) = u \) and the \( \log(\epsilon + u) \) functions, respectively. Then, we compare how many of the original outliers are among the 50 highest reconstruction values. The proposed method using the log function proved to be the most robust against outliers, with 47 from the 50 true outliers detected. It is followed by the variant with the identity function, which identified 27 outliers, and finally by K-SVD, which was naturally not able to identify any of the original outliers. This example also shows that concavity of \( g \) helps in better identifying outliers.
The classifier uses a Bag of Visual Words (BoVW) approach state-of-the-art methods such as K-SVD and LC-KSVD [18].

Experiments on synthetic and real world datasets show that our approach is able to achieve higher quality dictionaries which better generalize data. Table 1 shows the overall accuracies. These experiments clearly demonstrate that the quality of the dictionaries computed by the proposed robust dictionary learning method is indeed superior even to methods that uses labels for dictionary learning [18].

4. CONCLUSIONS

In this work, we proposed a generic dictionary learning framework which takes advantage of a composition of two concave functions to generate robust dictionaries with very little outlier interference. We also came up with a heuristic initialization which can further increase the identification of outliers, through the use of undercomplete dictionaries. Experiments on synthetic and real world datasets show that the proposed methods outperform some of the state-of-the-art methods such as K-SVD and LC-KSVD, since our approaches are able to achieve higher quality dictionaries which better generalize data.
5. REFERENCES


