BEYOND L2-LOSS FUNCTIONS FOR LEARNING SPARSE MODELS

Karthikeyan Natesan Ramamurthy[+], *Aleksandr Y. Aravkin*[‡], *Jayaraman J. Thiagarajan*[†]

[+]IBM Thomas J. Watson Research Center, Yorktown Heights, NY
[‡]Department of Applied Mathematics, University of Washington, Seattle, WA[*]
[†]Lawrence Livermore National Laboratory, Livermore, CA

ABSTRACT

In sparse learning, the squared Euclidean distance is a popular choice for measuring the approximation quality. However, the use of other forms of parametrized loss functions, including asymmetric losses, has generated research interest. In this paper, we perform sparse learning using a broad class of smooth piecewise linear quadratic (PLQ) loss functions, including robust and asymmetric losses that are adaptable to many real-world scenarios. The proposed framework also supports heterogeneous data modeling by allowing different PLQ penalties for different blocks of residual vectors (*split*-PLQ). We demonstrate the impact of the proposed sparse learning in image recovery, and apply the proposed split-PLQ loss approach to tag refinement for image annotation and retrieval.

Index Terms— PLQ, regularization, heterogeneous data, robust loss, tag refinement

1. INTRODUCTION

Deriving predictive inference from data requires both modeling the generating process, and estimating model parameters from input data. We consider the case of the widely-used linear model. The high-dimensional observation vector, $y \in \mathbb{R}^M$, can be approximated using a linear combination of representative columns in the *dictionary* matrix $D \in \mathbb{R}^{M \times K}$. The complexity of the linear model can be reduced by shrinking the small entries in a to zero [1]. The *sparse code* vector a can be optimized as

$$\min_a \rho_1(y - Da) + \lambda \rho_2(a) \tag{1}$$

where ρ_1 is the *loss function* that acts on the residual $r := y - Da$, ρ_2 is the sparsity *regularizer*, and λ is the regularization penalty that controls the trade-off between loss and regularization. The choice of loss function ρ_1 affects the measure of deviation between the observed and predicted data. Furthermore, the dictionary D can be adapted from the observations, $\{y_i\}_{i=1}^T$, when T is sufficiently large, by jointly minimizing the sum of T objectives given by (1) over D and $\{a\}_{i=1}^T$. Sparse coding and dictionary learning have widespread applications in speech and audio processing [2], image analysis and recovery, compressive sampling [3], unsupervised, supervised, semi-supervised, and transfer learning [4]. However most existing dictionary learning algorithms [5] are customized to the case where ρ_1 is the ℓ_2 loss function, which is equivalent to assuming a Gaussian distribution for the residual between observed and predicted data. When data are contaminated by outliers, robust loss functions can significantly improve performance relative to ℓ_2. Common applications of robust losses include learning econometric models that can tolerate a small fraction of bad years to the company, and processing images where a few pixels are corrupted due to saturation noise from sensors.

In this paper, we develop a flexible dictionary learning and sparse coding framework, allowing ρ_1 to be a member of a rich class of functions suitable for many real-world challenges. This class includes penalties that are: (a) robust to outliers, (b) block-assignable or *split*, acting differentially on specified subvectors of r, and (c) asymmetric, allowing differential treatment of positive and negative elements of r. All of these goals can be achieved by considering the general class of piecewise linear quadratic (PLQ) penalties [6, Definition 10.20], which comprise convex penalties whose domain can be represented as the union of finitely many polyhedral sets, relative to which the penalty can be expressed as a general (convex) quadratic. PLQ penalties include robust penalties such as ℓ_1, Huber, and Vapnik, asymmetric penalties such as quantile [7], and quantile Huber [8], as well as the classical ℓ_2 penalty (Figure 1).

2. ALGORITHMIC FORMULATION

We begin by formulating a generalized batch dictionary learning problem:

$$\begin{aligned} \min_{A,D} \quad & \rho_1(Y - DA) + \rho_2(A) + \rho_3(A) \\ & \text{subject to } A \in \mathcal{A}, D \in \mathcal{D}. \end{aligned} \tag{2}$$

where $Y = [y_1 y_2 \dots y_T]$ is the observation matrix, $A = [a_1 a_2 \dots a_T]$ is the corresponding sparse code matrix, ρ_1 is

[*]Washington Research Foundation Fund for Innovation in Data-Intensive Discovery

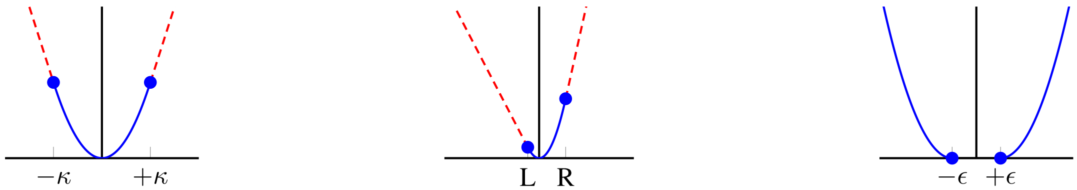


Fig. 1: Examples of smooth PLQ penalties for dictionary learning, from left to right: Huber, quantile Huber (0.3) [8], smooth insensitive loss.

the misfit loss function ρ_2 is the sparsity regularization, and ρ_3 encodes other prior information about the codes (for example, graph structure). The constraints $A \in \mathcal{A}$ and $D \in \mathcal{D}$ allow us to encode other prior information about the codes and the dictionary; for example, the columns of the dictionary may be normalized, and codes may be non-negative.

This problem is nonconvex, and is typically solved using block-coordinate descent or variants: dictionary D and codes A are updated in turn, with the other held fixed, using the *dictionary update* and *code update* steps. In this paper, we propose a modeling framework and optimization scheme to solve the general dictionary learning problem in (2), with simple constraints on A and D. Specifically, we allow ρ_1 to come from the class of smooth PLQ penalties, or a mixture of several PLQ penalties. A broad subclass of these penalties can be given a natural statistical interpretation, and their conjugate representation allows efficient optimization, enabling rapid prototyping [9], and including simple constraints $A \in \mathcal{A}$ [10]. We use this method to solve the code update problem.

2.1. Piecewise Linear-Quadratic penalties

We briefly review the PLQ penalties [9]. Every penalty in this class can be written as a convex conjugate to a quadratic function on a polyhedral set.

Definition 2.1. *A PLQ function is any function $\rho(U, M, b, B; \cdot)$ mapping from $\mathbb{R}^n$ to $\overline{\mathbb{R}} = \mathbb{R} \cup \infty$ having the representation*

$$\rho(C, c, M, b, B; y) = \sup_{Cu \le c} \left\{ u^T(b + By) - u^T M u \right\} \ , \quad (3)$$

where $M \in \mathcal{S}^n_+$ the set of real symmetric positive semidefinite matrices, $c \in \mathbb{R}^k$, $b, u \in \mathbb{R}^m$, $C \in \mathbb{R}^{k \times m}$, and $b + By$ is an injective affine transformation in y, with $B \in \mathbb{R}^{m \times n}$.

Any PLQ formulation can be optimized using an interior point method together with the representation in (3), and polyhedral constraints on y can also be included [10]. The PLQ representation allows a calculus that captures key modeling operations. For example, the sum of two PLQ penalties is also a PLQ penalty, and PLQ penalties are closed under affine composition [9]. We present the following simple and practically useful lemma, showing that PLQ penalties can be easily defined over a product space.

Lemma 2.2 (Product action)**.** *A PLQ $\rho(y) = \rho_1(y_1) + \rho_2(y_2)$, where y_1 and y_2 are sub-blocks of the vector y, is easily written in terms of addition and affine composition; namely*

$$\rho(y) = \rho_1(M_1 y) + \rho_2(M_2 y),$$

where $M_1 y = y_1$ and $M_2 y = y_2$.

2.2. Block coordinate descent

For the full nonconvex problem (2), the natural approach is to alternate between updating sparse codes A and the dictionary D, which is an instance of block coordinate descent. When the penalties ρ_1, ρ_2 and ρ_3 are smooth, standard convergence results for block coordinate descent can be obtained with e.g. [11, Proposition 2.7.1]. However, ρ_2 is generally non-smooth (ℓ_1 norm) and we are also interested in a general theory that applies to the entire PLQ class. Block-coordinate descent for a class of problems general enough to accommodate our framework is studied in [12], but it has a sharp condition for convergence that requires ρ_1 to be smooth.

Theorem 2.3. *Suppose that ρ_1 in* (2) *is differentiable, ρ_2 and ρ_3 are convex, and the sets $\mathcal{D}$ and $\mathcal{A}$ are convex. Then block coordinate descent (alternating minimization in A and D) converges to a stationary point of* (2)*.*

Proof. By assumption, ρ_1 is differentiable on its effective domain; furthermore, the entire objective is convex in A. By [12, Lemma 3.1 and Theorem 4.1(b)], every cluster point of the sequence generated from block-coordinate descent is a stationary point of (2). $\square$

From the application perspective, the requirement that ρ_1 be smooth is not particularly limiting, and in fact, in sparse high dimensional regression the smoothed version of the quantile penalty called the *quantile Huber* has been shown to outperform the standard quantile penalty [8]. Further, since PLQ penalties are closed under Moreau-Yosida smoothing [13, Proposition 4.11], any PLQ penalty candidate for ρ_1 can be smoothed and it will still be in the PLQ class.

2.3. Dictionary update problem

In this section, we show how to solve for $D = \operatorname{argmin} \rho_1(Y - DA)$, for a fixed set of sparse codes A, and prove the convergence of our scheme. In the least squares case, it is

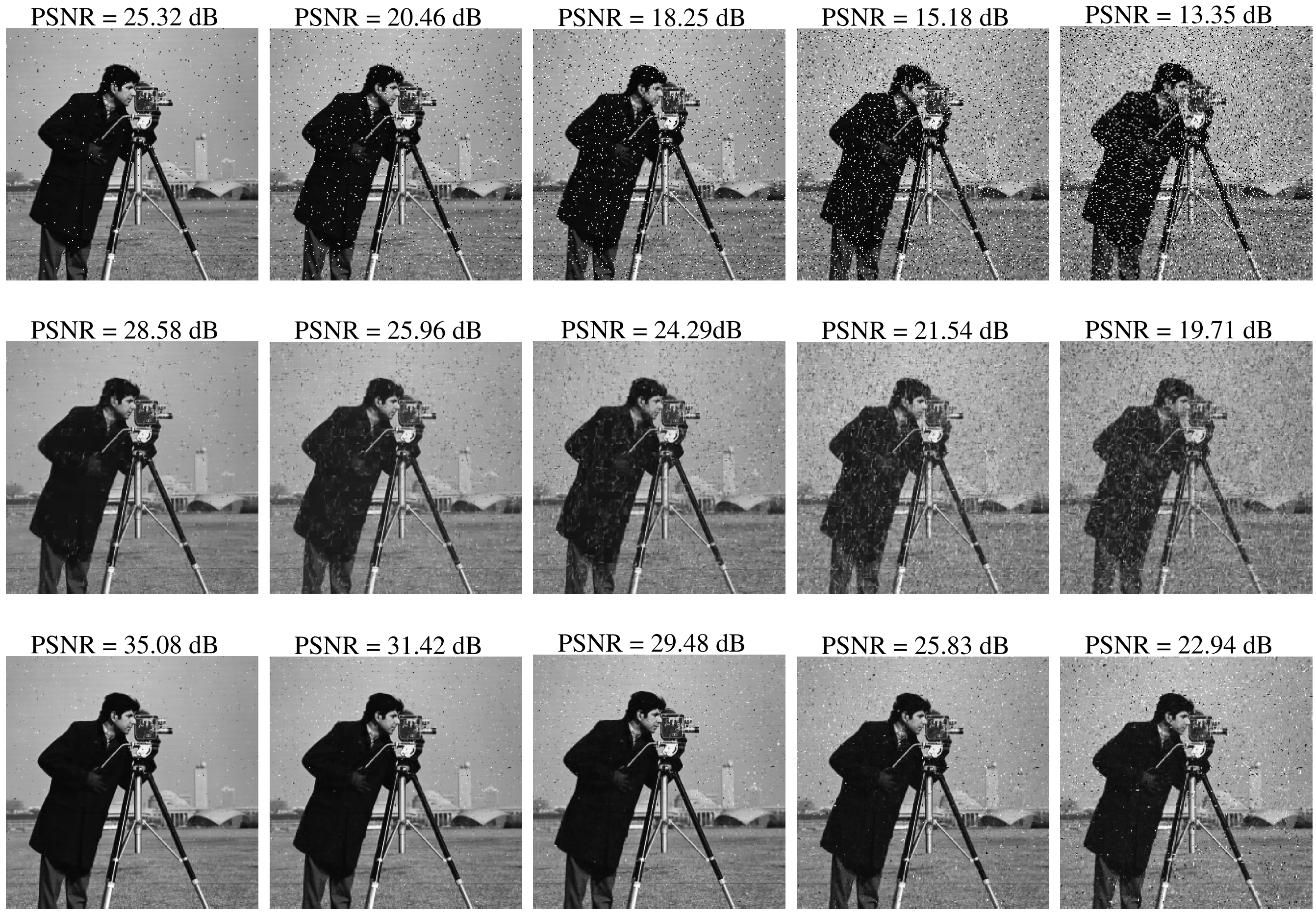


Fig. 2: Robust Image Modeling - Row 1 shows images corrupted by increasing levels of salt and pepper noise. Rows 2 and 3 show the images recovered using sparse models learned with the ℓ_2 and Huber penalties, respectively.

straightforward to implement a block-coordinate optimization scheme on the columns of D, obtaining closed-form updates as we loop over the columns.

In the general case, let us suppose that we wish to update the j-th column of D_j. Letting a_j denote the jth *row* of A, d_j denote the jth column of D, and $D_{/j}$ to denote the dictionary with the jth column deleted, it is easy to see that

$$DA - D_{/j}A = d_j a_j^T.$$

For penalties ρ_1 which decompose over the columns of the residual $Y - DA$, the optimization formulation to estimate d_j is given by

$$\bar{d}_j = \operatorname*{argmin}_{d} \rho_1(Y_j - d a_j^T). \tag{4}$$

with $Y_j = Y - D_{/j}A$. For the least-squares case, this update problem has a closed form solution; and in the general case, the structure of the problem is very simple: the kth entry of $d_{j,k}$ is determined by solving a scalar optimization problem

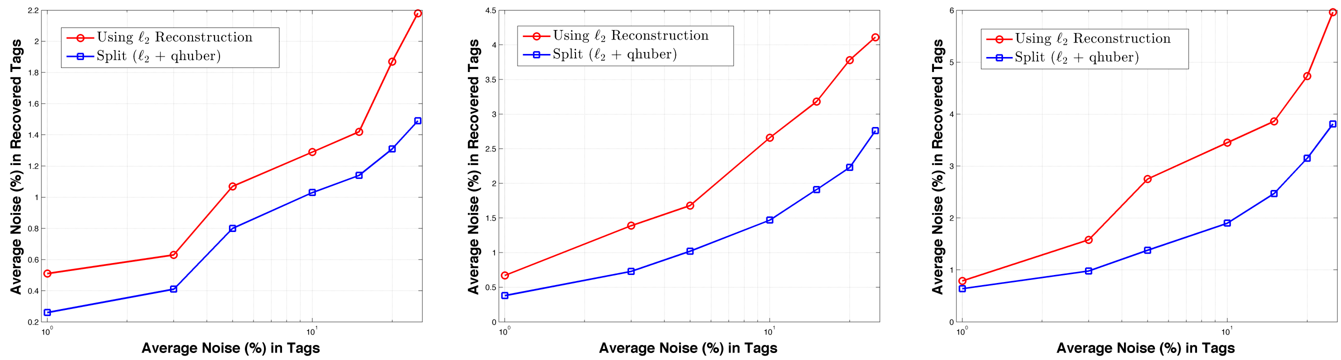
$$d_{j,k} = \operatorname*{argmin}_{d_k} \rho_1(Y_j(k,\cdot) - d_k a_j).$$

Since this is a 1-dimensional optimization problem, the Barzilai-Borwein [14] line search method is equivalent to Newton's method in the quadratic case (after 2 steps). Motivated by this, we use the limited-memory BFGS (L-BFGS) with Barzilai-Borwein line search to solve (4). For quadratic ρ_2, this method converges in two iterations per column, as expected, and for general smooth ρ_2, such as the Huber, it is also rapidly convergent. Since Theorem 2.3 requires ρ_1 to be smooth, block-column coordinate descent converges by [11, Proposition 2.7.1].

3. EXPERIMENTS

3.1. Image Recovery Using Robust Dictionary Learning

In this experiment, we consider the problem of recovering images corrupted by an additive noise. In such scenarios, a generalizable model should ignore the underlying noise, and describe only the relevant patterns in the image. When the noise is Gaussian, the traditional sparse models with ℓ_2 loss function, can be very effective in discovering patterns. How-



(a) (b) (c)

Fig. 3: Tag Refinement using ℓ_2 (red) and mixed ℓ_2-Huber penalties (blue) for: (a) 0%, (b) 5%, and (c) 10% training noise levels. Using appropriate robust penalties for the tags result in improved recovery performance at all noise levels.

ever, when the noise model is non-Gaussian, the sparse model learned using this procedure will no longer be robust. In our setup, the images are corrupted by salt-and-pepper noise, which manifests as randomly occurring white and black pixels in the image. We propose to use the Huber penalty as the loss function, since it can learn median patterns in the dictionary, thereby resulting in a robust model.

Given an image I, we extract non-overlapping patches of size 8×8 and vectorize these patches into a matrix denoted X. We vary the level of salt and pepper noise from 1% to 15%. We learn dictionaries using different penalties, and compare the reconstruction obtained using the learned sparse model with the original clean image (based on PSNR). When the model is robust, we expect that the noise will not be a part of the dictionary elements, and hence the reconstruction will be of high quality. Note that we do not perform any explicit denoising, and only evaluate the quality of the reconstruction from the model. From the results in Figure 2 the robustness of the Huber penalty is clearly evident.

3.2. Refining Tags for Image Retrieval

Textual descriptors, or *tags*, are useful meta-data for images in retrieval applications. The goal of automatic image annotation is to predict new tags, and possibly refine existing noisy tags, based on information from visually similar images. In this experiment, we will consider the problem of refining the noisy tags of a novel image using a set of training images.

Given a set of training images, we use the *Gist* features [17] to describe the visual content. The set of visual features are stored in the matrix X, and their corresponding textual descriptors are stored in the matrix B. For each image, a tag vector is typically a binary vector that indicates the relevance of each semantic topic from a pre-defined vocabulary. Given a novel image feature y, and its noisy tag vector h, our goal is to obtain a refined estimate $\bar{h}$. We propose to exploit the correlations between the features and tags, using sparse coding, to perform tag refinement. Using the set of training examples, we construct the dictionary $D = [X^T \ \gamma B^T]^T$, where γ is the scaling factor used to balance the total energy of features and tags. Similarly, the test sample is described as $z = [y^T \ \gamma h^T]^T$. By assuming that the features and tags are clustered along subspaces, this structure can be discovered using sparse coding on examples: $\min_a \|z - Da\|_2^2 + \lambda \|a\|_1$. The refined tag vector can then be estimated as $\bar{h} = Ba$. This formulation assumes that both features and semantic descriptors can be recovered using the same set of sparse coefficients. However, the ℓ_2 penalty is not robust, and thus unsuitable for measuring the misfit in the reconstruction of tag vectors. To improve the recovery, we use different penalties for modeling visual features and tag vectors:

$$\min_a \rho_1^{(1)}(y - Xa) + \rho_1^{(2)}(h - Ba) + \lambda \|a\|_1, \tag{5}$$

where $\rho_1^{(1)}$ is the ℓ_2 penalty, and $\rho_1^{(2)}$ is the Huber penalty.

For our experiment, we used the Corel-5K data set [18], which contains $5,000$ images in total, and each image is annotated with 1 to 5 keywords. We used $4,500$ images as training data, and evaluated the performance using the rest. The total number of keywords in the vocabulary is 260. We varied the level of noise in the test tags, by randomly flipping $\{1\%, 3\%, 5\%, 10\%, 15\%, 20\%, 25\%\}$ of the entries in each binary tag vector. We estimated the refined tags, and computed the average noise $(\%)$ in the refined tag vectors. Figure 3(a) plots the performance obtained using the ℓ_2 penalty for the entire residual, and the mixed ℓ_2-Huber penalty. Clearly, the robust variant using the mixed penalty provides improved recovery at all noise levels. Furthermore, we corrupted the tag vectors of the training data also with different levels of noise and studied the performance deterioration (Figures 3(b) and (c)). We found that using mixed penalties provided superior performance in all cases.

4. REFERENCES

[1] Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani, *The elements of statistical learning*, Springer, 2009.

[2] Daniele Giacobello, Mads Græsbøll Christensen, Manohar N Murthi, Søren Holdt Jensen, and Marc Moonen, "Sparse linear prediction and its applications to speech processing," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 5, pp. 1644–1657, 2012.

[3] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer, 2010.

[4] Jayaraman J Thiagarajan, Karthikeyan Natesan Ramamurthy, Pavan Turaga, and Andreas Spanias, "Image understanding using sparse representations," *Synthesis Lectures on Image, Video, and Multimedia Processing*, vol. 7, no. 1, pp. 1–118, 2014.

[5] Ivana Tosic and Pascal Frossard, "Dictionary learning," *Signal Processing Magazine, IEEE*, vol. 28, no. 2, pp. 27–38, 2011.

[6] R.T. Rockafellar and R.J.B. Wets, *Variational Analysis*, vol. 317, Springer, 1998.

[7] Ichiro Takeuchi, Quoc V Le, Timothy D Sears, and Alexander J Smola, "Nonparametric quantile estimation," *The Journal of Machine Learning Research*, vol. 7, pp. 1231–1264, 2006.

[8] Aleksandr Aravkin, Aurelie Lozano, Ronny Luss, and Prabhajan Kambadur, "Orthogonal matching pursuit for sparse quantile regression," in *Data Mining (ICDM), 2014 IEEE International Conference on*. IEEE, 2014, pp. 11–19.

[9] Aleksandr Y. Aravkin, James V. Burke, and Gianluigi Pillonetto, "Sparse/robust estimation and kalman smoothing with nonsmooth log-concave densities: Modeling, computation, and theory," *Journal of Machine Learning Research*, vol. 14, pp. 2689–2728, 2013.

[10] Aleksandr Y. Aravkin, James V. Burke, and Gianluigi Pillonetto, "Linear system identification using stable spline kernels and plq penalties," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*. IEEE, 2013, pp. 5168–5173.

[11] Dimitri P Bertsekas, *Nonlinear Programming*, Athena Scientific, second edition, 1999.

[12] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *Journal of Optimization Theory and Applications*, vol. 109, no. 3, pp. 475–494, 2001.

[13] James V. Burke and Tim Hoheisel, "Epi-convergent smoothing with applications to convex composite functions.," *SIAM Journal on Optimization*, vol. 23, no. 3, pp. 1457–1479, 2013.

[14] Jonathan Barzilai and Jonathan M. Borwein, "Two-point step size gradient methods," *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 1988.

[15] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *IEEE CVPR*, Jun. 2010, pp. 3501 –3508.

[16] Bin Cheng, Jianchao Yang, Shuicheng Yan, Yun Fu, and Thomas S Huang, "Learning with l1-graph for image analysis.," *IEEE transactions on image processing*, vol. 19, no. 4, pp. 858–66, apr 2010.

[17] Aude Oliva and Antonio Torralba, "Building the gist of a scene: The role of global image features in recognition," *Progress in brain research*, vol. 155, pp. 23–36, 2006.

[18] Pinar Duygulu, Kobus Barnard, Joao FG de Freitas, and David A Forsyth, "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary," in *Computer VisionECCV 2002*, pp. 97–112. Springer, 2002.