Extending Kernel PCA through Dualization: Sparsity, Robustness and Fast Algorithms

Francesco Tonin [*1] **Alex Lambert** [*1] **Panagiotis Patrinos** [1] **Johan A.K. Suykens** [1]

Abstract

The goal of this paper is to revisit Kernel Principal Component Analysis (KPCA) through dualization of a difference of convex functions. This allows to naturally extend KPCA to multiple objective functions and leads to efficient gradient-based algorithms avoiding the expensive SVD of the Gram matrix. Particularly, we consider objective functions that can be written as Moreau envelopes, demonstrating how to promote robustness and sparsity within the same framework. The proposed method is evaluated on synthetic and real-world benchmarks, showing significant speedup in KPCA training time as well as highlighting the benefits in terms of robustness and sparsity.

1. Introduction

Kernel Principal Components Analysis (KPCA, Schölkopf et al. (1998)) stands as one of the most widely used tools for unsupervised learning, with applications to dimensionality reduction, denoising, or features extraction. By embedding the data in some higher dimensional space thanks to a feature map, KPCA aims at finding orthogonal directions in the feature space that allow to best reconstruct the empirical covariance operator.

The classical way to solve the KPCA problem is to compute the singular values decomposition (SVD) of the Gram matrix, a costly $\mathcal{O}(n^3)$ operation whose complexity scales cubicly with respect to the number n of datapoints. This prevents KPCA from being used in large-scale scenarios, and, while some solutions exist, they mostly boil down to subsampling in fixed-size schemes to approximate the nonlinear mapping (Langone & Suykens, 2017) or focus on the online learning setting (Günter et al., 2007; Chin & Suter, 2007; Honeine, 2012). Works on speeding up the simpler problem of linear PCA exist (see (Gemp et al., 2021) and references therein), but they cannot deal with infinite-dimensional feature spaces.

From an optimization standpoint, the KPCA problem can be formulated as variance maximization under orthonormality constraints. Such problem belongs to the wider family of differences of convex functions (DC) problems, which has received a great deal of attention in multiple applications (e.g., see (Tao & An, 1997)). In particular, in (Beck & Teboulle, 2021), PCA was investigated as a DC problem, but only in the case of linear PCA, i.e., not considering (potentially infinite-dimensional) feature mappings, and only for the simpler problem of finding the first component where the orthogonality constraints become void.

Enforcing desirable properties to the solution, such as sparsity or robustness, is a long-standing open problem in KPCA. While many works have investigated robust/sparse KPCA (e.g., robust KPCA in (Nguyen & Torre, 2008; Kim & Klabjan, 2020; Wang & Tanaka, 2020a; Fan & Chow, 2020) and sparse KPCA in (Wang & Tanaka, 2016; Guo et al., 2019b; Tipping, 2000; Smola et al., 2002)), they use several different ad-hoc approaches or heuristics such as weighting schemes (Alzate & Suykens, 2008), leading to a multitude of different optimization problems. In (Thiao et al., 2010), a DC program for the specific problem of sparse linear PCA was explained, but is not extended to also handle robust losses within the same framework; notably, it does not deal with nonlinear feature maps nor with more than one component. The idea of using infimal convolution to design sparse or robust losses is exploited notably in Sangnier et al. (2017); Laforgue et al. (2020) for regression problems, and is known to work well in duality settings.

In this paper, we derive a general dual-based formulation for KPCA leading to a difference of convex function objective which encompasses both the variance and robust/sparse objective functions in the same framework. We derive efficient optimization algorithms showing significant speedups compared to the standard SVD solvers for KPCA. In particular, our approach allows to solve *infinite*-dimensional KPCA problems in the dual. We focus on objectives that can be written as Moreau envelopes, which include the Huber and ϵ-insensitive losses, inducing robustness and spar-

[*]Equal contribution [1]ESAT-STADIUS, KU Leuven, Belgium. Correspondence to: Francesco Tonin <francesco.tonin@esat.kuleuven.be>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

sity, respectively. We show how the resulting optimization problems can be tackled with efficient algorithms for each objective.

The paper is structured as follows. In Section 2, we formulate the general KPCA problem as a difference of convex functions, which leads to a flexible framework that can be extended to other loss functions beyond the square loss. We present in Section 3 a gradient-based optimization algorithm able to efficiently solve the standard KPCA problem through dualization. Later, in Section 4 we exploit the flexibility of the proposed DC framework by modifying the objective function to promote robustness and sparsity. Finally, numerical experiments on both synthetic and real-world benchmarks are presented in Section 5, showing faster training times and illustrating the promotion of sparsity and robustness induced in the solution. All proofs are deferred to the appendix.

2. Problem Formulation

Notation: Given a symmetric real matrix $M \in \mathbb{R}^{n\times n}$, $\lambda(M) \in \mathbb{R}^n$ is the vector of its eigenvalues ordered decreasingly. For a linear operator Γ between Hilbert spaces, $\Gamma^\sharp$ denotes its adjoint. When $w \in \mathcal{H}$ is a vector, $w^\sharp$ refers to the linear form $x \mapsto \langle w, x\rangle$. I_s is the identity matrix of size $s\times s$. $\phi\colon \mathcal{X} \to \mathcal{H}$ is a mapping to a Hilbert space $\mathcal{H}$. ϕ induces a positive definite kernel function $k\colon \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ with associated RKHS $\mathcal{H}_k$. Given a vector $W = (w_1, \dots, w_s) \in \mathcal{H}^s$, we denote by $\mathcal{G}(W) \in \mathbb{R}^{s\times s}$ the Gram matrix such that $\mathcal{G}(W)_{ij} = \langle w_i, w_j\rangle$. $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm. For a convex set $\mathcal{C}$, $\iota_{\mathcal{C}}(\cdot)$ is its indicator function: 0 on $\mathcal{C}$ and $+\infty$ otherwise. The infimal convolution is denoted $\square$, and the Fenchel-Legendre transform of a function f is $f^\star$.

Let $\mathcal{X}$ be some input space, and $(x_i)_{i=1}^n \in \mathcal{X}^n$ n datapoints in it. These datapoints are embedded in a feature space $\mathcal{H}$ by means of a feature map $\phi\colon \mathcal{X} \to \mathcal{H}$. Up to some recentering of ϕ, we assume that $\sum_{i=1}^n \phi(x_i) = 0$. The KPCA problem consists in finding orthogonal directions in the feature space that allow for the best low rank approximation of some empirical version of the covariance operator

$$\Sigma \coloneqq \frac{1}{n}\sum_{i=1}^n \phi(x_i)\phi(x_i)^\sharp \in \mathcal{L}(\mathcal{H}).$$

Given a desired number of components s, KPCA can be reformulated as finding s directions in the Hilbert space $\mathcal{H}$ that maximize the variance under orthonormal conditions. When $\mathcal{H}$ is some finite dimensional $\mathbb{R}^p$ space, and denoting $\Phi \in \mathbb{R}^{n\times p}$ the row-wise concatenations of the $[\phi(x_i)]_{i=1}^n$, this reduces to

$$\sup_{W\in\mathbb{R}^{p\times s}} \|\Phi W\|_{\mathrm{F}}^2 \text{ s.t. } W^\top W = I_s.$$

The Stiefel manifold over Hilbert spaces: Given a feature space $\mathcal{H}$ and a positive integer s, we introduce the Stiefel manifold of orthonormal s-frames in $\mathcal{H}$ as

$$\mathcal{S}_{\mathcal{H}}^s \coloneqq \{W \in \mathcal{H}^s \mid \mathcal{G}(W) = I_s\}.$$

When $\mathcal{H} = \mathbb{R}^d$ is endowed with the Euclidean scalar product, the Stiefel manifold corresponds to the usual set of matrices $\mathcal{S}_{\mathbb{R}^d}^s = \left\{M \in \mathcal{M}_{d,s}(\mathbb{R}) | M^\top M = I_s\right\}$. We define $\Gamma\colon \mathcal{H}^s \to \mathbb{R}^{n\times s}$ as the linear operator such that for all $(i,j) \in [n\times s]$ and $W = (w_1, \dots, w_s) \in \mathcal{H}^s$, $[\Gamma W]_{ij} = \langle \phi(x_i), w_j\rangle$. The KPCA problem then reduces to

$$\sup_{W\in\mathcal{S}_{\mathcal{H}}^s} \frac{1}{2}\|\Gamma W\|_{\mathrm{F}}^2. \tag{1}$$

While the constraint $W \in \mathcal{S}_{\mathcal{H}}^s$ is not convex, it can be relaxed to the convex hull of the Stiefel manifold as the solutions necessarily lie on the boundary (Uschmajew (2010), Lemma 2.7) which justifies the fact that we consider the constraint convex in what follows.

Kernel PCA from SVD: The usual way to solve the KPCA problem is to express the directions $W = [w_j]_{j=1}^s$ as linear combinations of the features, introducing coefficients $(\alpha_{ij})_{i,j=1}^{n,s} \in \mathbb{R}^{n\times s}$ such that

$$w_j = \sum_{i=1}^n \alpha_{ij}\phi(x_i).$$

Quick algebraic manipulations then ensure that the solution to Problem 1 can be obtained by taking the coefficients α to be the top-s eigenvectors of the Gram matrix $G = [k(x_i, x_j)]_{i,j=1}^n$, rescaled so that the directions have unit norm. Thus KPCA is solved by performing the SVD of some $n\times n$ matrix, an operation scaling as $\mathcal{O}(n^3)$ and quite slow even for moderate sizes of datasets. Throughout this paper we make the assumption that G is full rank, which typically happens when the feature space is infinite dimensional (*e.g.* Gaussian kernel) and the data does not contain any duplicate. This assumption is critical to the derivation of the gradient in Section 3.

Difference of convex functions: The KPCA optimization problem can be cast into the minimization of a difference of convex functions of the form

$$\inf_{W\in\mathcal{H}^s} g(W) - f(\Gamma W). \tag{2}$$

Indeed, plugging back $f = \frac{1}{2}\|\cdot\|_{\mathrm{F}}^2$ and $g = \iota_{\mathcal{S}_{\mathcal{H}}^s}(\cdot)$ into Problem 2 one gets back the original formulation from Problem 1. One of the advantages of this formulation is that it becomes possible to slightly modify the loss function f so as to enforce specific properties such as robustness or sparsity for the solution.

The goal of this work is two-fold:

- Exploit gradient-based optimization schemes to solve KPCA without using the SVD of G, resulting in computational advantages (see Section 3).
- Extend the KPCA problem to a larger set of objective functions, with the benefit of enforcing robust or sparse solutions (see Section 4).

3. Solving Kernel PCA with Gradient Descent

In this section, we use duality principles to derive suitable optimization algorithms to solve the KPCA problem introduced in Problem 1. We begin by a general proposition performing the dualization of problems that can be written as the minimization of a difference of two convex functions. The proof is similar to Toland (1979), with an additional care given to handling the operator Γ.

Proposition 3.1 (Dual of difference of convex functions)**.** *Let $\mathcal{U}, \mathcal{K}$ be two Hilbert spaces, $g\colon \mathcal{U} \to \bar{\mathbb{R}}$ and $f\colon \mathcal{K} \to \bar{\mathbb{R}}$ be two convex lower semi-continuous functions and $\Gamma \in \mathcal{L}(\mathcal{U}, \mathcal{K})$. The problem*

$$\inf_{W \in \mathcal{U}} \; g(W) - f(\Gamma W)$$

admits the dual formulation

$$\inf_{H \in \mathcal{K}} \; f^\star(H) - g^\star(\Gamma^\sharp H), \tag{3}$$

and strong duality holds.

The main motivation for going from the primal problem to the dual in the KPCA case is that the dual variable H is a finite dimensional matrix, suitable to gradient-based optimization schemes. However this comes at the cost of having to handle the term $g^\star(\Gamma^\sharp H)$ which encodes information related to the Stiefel manifold. In Section 3.1, we show that this term is related to the nuclear norm of some low dimensional matrix and derive a gradient for it, before exploring in Section 3.2 the link between the critical points of the dual function and those of the reconstruction cost associated to the Gram matrix. Finally in Section 3.3, we show how to compute the projections associated to new points without having to compute the primal solution.

3.1. Nuclear Norm Gradient Computation

The following proposition instantiates the term $g^\star(\Gamma^\sharp H)$ for the KPCA problem.

Proposition 3.2. *Let g be the indicator function of the Stiefel manifold and Γ as in Problem 1. Then for all $H \in \mathbb{R}^{n \times s}$,*

$$g^\star(\Gamma^\sharp H) = \operatorname{Tr} \sqrt{H^\top G H} \tag{4}$$

We recognize in Equation (4) the nuclear norm of the matrix $\sqrt{H^\top G H}$, which is well-defined since G is a Gram matrix associated to a positive definite kernel.

Remark 3.3. In the proof, we first show that $g^\star(\Gamma^\sharp H) = \left\| G^{\frac{1}{2}} H \right\|_{S_1}$ where $\|\cdot\|_{S_1}$ is the Schatten 1-norm, *i.e.* the nuclear norm. While the dependency in H makes this easier to handle from an optimization standpoint, the dependency in $G^{\frac{1}{2}}$ would require to perform the SVD that we want to avoid, leaving us to work with $\operatorname{Tr} \sqrt{H^\top G H}$.

In what follows, we define $\pi(H) := \operatorname{Tr} \sqrt{H^\top G H}$ and build on Lewis (1996) to give a gradient expression for π used in subsequent optimization algorithms.

Proposition 3.4. *If all eigenvalues of $H^\top G H$ are positive then π is differentiable at H with gradient*

$$\nabla \pi(H) = G H U^\top \operatorname{diag}\left(\frac{1}{\sqrt{\lambda(H^\top G H)}} \right) U, \tag{5}$$

where $U \in \mathbb{R}^{s \times s}$ is an orthogonal matrix satisfying $H^\top G H = U^\top \operatorname{diag}\left(\lambda(H^\top G H)\right) U$.

To compute the gradient given in Equation (5), one needs to perform the SVD of the matrix $H^\top G H \in \mathbb{R}^{s \times s}$. In the context of a (relatively) small number of components, this SVD is computationally cheap. This motivates the use of gradient-based method to obtain the dual solution faster than by exploiting the SVD of the $n \times n$ matrix G.

More precisely, the computational complexity associated to the computation of this gradient can be bounded by the sum of

- Computation of $H^\top G H$ in $\mathcal{O}(sn^2)$
- SVD of $H^\top G H$ in $\mathcal{O}(s^3)$
- Computation of Equation (5) in $\mathcal{O}(ns^2 + s^3)$ by reusing the precomputed GH.

Remark 3.5. Due to the dependency of the gradient in $\frac{1}{\sqrt{\lambda(H^\top G H)}}$, it is not Lipschitz continuous in H, which prevents the use of fixed stepsizes schemes in classical optimization algorithms.

3.2. Critical Points of the Dual Problem

Overall, the dual problem to KPCA reads

$$\inf_{H \in \mathbb{R}^{n \times s}} \frac{1}{2} \operatorname{Tr}(H^\top H) - \pi(H). \tag{6}$$

Solving this nonconvex problem can be challenging, as local minima or saddle points phenomena can occur. Typically, algorithms such as the DC algorithm (Tao & An, 1997) ensure convergence towards a critical point, that is a point H at which the gradient is nullified. In the following proposition, we explicit the link between critical points of the dual function and those of the reconstruction cost associated to the Gram matrix.

Proposition 3.6. *Let $J(H) = \frac{1}{4}\left\|G - HH^\top\right\|_{\mathrm{F}}^2$. Let $\hat{H}$ be a critical point of Problem 6. Then $\nabla J(\hat{H}) = 0$.*

The characterization of the critical points of J is given in (Wright & Ma, 2022). We solve the dual problem (6) using the L-BFGS optimization algorithm, which in practice we observe always avoids sub-optimal critical points. More details on the optimization algorithm are given in Section 5.

3.3. Exploiting the Dual Solution

Assume that solving Problem 6 has produced an optimal dual solution that we denote $\hat{H}$. In general, finding the corresponding directions in the feature space (primal variables) involves solving the optimization problem

$$\hat{W} \in \operatorname*{arg\,min}_{W \in \mathcal{H}^s} \; g(W) - \langle \hat{H}, \Gamma W \rangle. \tag{7}$$

This optimization problem is non-trivial when we do not have more information about the solution $\hat{H}$. However, when $\hat{H}$ is obtained by the SVD of G, there is a linear dependency between the directions in the feature space and the dual variables. This is exemplified in the following proposition where we show that the SVD of G allows to pick an optimal dual solution.

Proposition 3.7. *Let $G = U^\top \Sigma U$ be the SVD of G with $U \in \mathbb{R}^{n\times n}$ being orthonormal and Σ being diagonal. Let $H^{svd} = U_s^\top \sqrt{\Sigma}_s$ where $U_s \in \mathbb{R}^{s\times n}$ gathers the top-s eigenvectors of G and $\Sigma_s \in \mathbb{R}^{s\times s}$ the top-s eigenvalues. Then H^{svd} is a solution to Problem 6.*

In our case, we do not have access to such an optimal solution and must find a way to compute the projections with the only knowledge of $\hat{H}$. It turns out that recovering the projection on the principal components is possible using the kernel trick, as proposed in the following.

Proposition 3.8. *Let $x \in \mathcal{X}$ and $G_x = [k(x, x_i)]_{i=1}^n \in \mathbb{R}^n$. Let $\hat{W} \in \mathcal{H}^s$ be a solution to Problem 1 and $\hat{H} \in \mathbb{R}^{n\times s}$ be a solution to Problem 6. The projections of $\phi(x)$ onto the principal components are given by*

$$[\langle \phi(x), \hat{w}_j \rangle]_{j=1}^s = G_x^\top \hat{H} U^\top \operatorname{diag}\left(\lambda(\hat{H}^\top G \hat{H})\right)^{-\frac{1}{2}} U \tag{8}$$

where U is obtained from the SVD of $\hat{H}^\top G \hat{H}$ as in Proposition 3.4.

Note that the expression in Equation (8) does not directly involves the directions in the feature space encoded in $\hat{W}$.

4. Beyond Variance Maximization

In this section, we propose to modify the variance objective used in the original KPCA problem to promote desirable properties such as sparsity or robustness in the dual variable H. The idea is based on using objectives obtained from infimal convolution with the squared norm, known as *Moreau envelopes* (Moreau, 1965). We focus on variance-like objectives of the form

$$f = \frac{1}{2}\left\|\cdot\right\|_{\mathrm{F}}^2 \square \, \Psi,$$

where $\Psi \colon \mathbb{R}^{n\times s} \to \bar{\mathbb{R}}$ is a well-chosen function enforcing desirable properties. Compatibility between the Fenchel-Legendre transform and the infimal convolution operator then allows to write the dual to Problem 2 as

$$\inf_{H \in \mathbb{R}^{n\times s}} \; \frac{1}{2}\left\|H\right\|_{\mathrm{F}}^2 + \Psi^\star(H) - \pi(H). \tag{9}$$

The section is organized as follows: we begin in Section 4.1 by expliciting some choices of Ψ that give rise to the Huber and ϵ-insensitive objectives (known to respectively promote robustness and sparsity) before exploiting their Moreau envelope structure to design dedicated optimization schemes in Section 4.2.

4.1. Huber and ϵ-insensitive Objectives

In what follows, let $\left\|\cdot\right\|$ be a norm on $\mathbb{R}^{n\times s}$ and $\left\|\cdot\right\|_\star$ be its dual norm. Given $t \geq 0$, the balls of radius t for these norms are respectively denoted as $\mathcal{B}_t$ and $\mathcal{B}_t^\star$.

Definition 4.1 (Huber)**.** Let $\kappa > 0$. The Huber objective with parameter κ is defined as

$$H_\kappa := \frac{1}{2}\left\|\cdot\right\|_{\mathrm{F}}^2 \square \, \kappa \left\|\cdot\right\|.$$

The Huber objective can be understood as the Moreau envelope of $\kappa\left\|\cdot\right\|$ with parameter 1. This case corresponds to $\Psi = \kappa\left\|\cdot\right\|$ and $\Psi^\star = \iota_{\mathcal{B}_\kappa^\star}$. The additional term in Problem 9 constrains the dual variable H to pertain to the ball of radius κ for the norm $\left\|\cdot\right\|_\star$, thus inducing robustess. While the choice $\left\|\cdot\right\| = \left\|\cdot\right\|_{\mathrm{F}}$ recovers the classical version of the Huber objective, it is to be noted that varying choices of norms allow to capture different notions of robustness (Lambert et al., 2022).

Definition 4.2 (ϵ-insensitive)**.** Let $\epsilon > 0$. The ϵ-insensitive objective with parameter ϵ is defined as

$$\ell_\epsilon := \frac{1}{2}\left\|\cdot\right\|_{\mathrm{F}}^2 \square \, \iota_{\mathcal{B}_\epsilon}.$$

The ϵ-insensitive objective is the Moreau envelope of some indicator function of a ball, corresponding to $\Psi = \iota_{\mathcal{B}_\epsilon}$ and $\Psi^\star = \epsilon\left\|\cdot\right\|_\star$. Depending on the choice of the norm, the additional term in Problem 9 can behave like a Lasso or group Lasso penalty, inducing sparsity in the iterates.

Table 1: Proximal operators for Huber and ϵ-insensitive objectives.

Objective	Ψ	$\Psi^\star$	$\mathrm{prox}_{\Psi^\star}(Y)$
H_κ	$\kappa \|\cdot\|$	$\iota_{\mathcal{B}^\star_\kappa}$	$\mathrm{Proj}_{\mathcal{B}^\star_\kappa}(Y)$
ℓ_ϵ	$\iota_{\mathcal{B}_\epsilon}$	$\epsilon \|\cdot\|_\star$	$Y - \mathrm{Proj}_{\mathcal{B}_\epsilon}(Y)$

Algorithm 1 DCA for Moreau envelope objectives

input : Gram matrix G
init : $H^{(0)} \in \mathbb{R}^{n\times s}, Y = 0$
for *epoch* t *from* 0 *to* $T-1$ **do**
  `// alternating gradient steps`
  $Y = \nabla\pi(H^{(t)})$ from Equation (5)
  $H^{(t+1)} = \mathrm{prox}_{\psi^\star}(Y)$
return $H^{(T)}$

4.2. Solving with DC Algorithm

We propose to solve Problem 9 using the well-known difference of convex functions algorithm (DCA) (Tao & An, 1997). The point of DCA is to search for a critical point of the problem, that is some $H \in \mathbb{R}^{n\times s}$ such that

$$\partial\pi(H) \cap \partial(\frac{1}{2}\|\cdot\|_{\mathrm{F}}^2 + \Psi^\star)(H) \neq \emptyset.$$

The search is performed sequentially in the subgradient of π, and in the subgradient of the Fenchel-Legendre transform of $\frac{1}{2}\|\cdot\|_{\mathrm{F}}^2 + \Psi^\star$, which is f. As f was chosen to be a Moreau envelope, its gradient is always defined as for all $Y \in \mathbb{R}^{n\times s}$,

$$\nabla\left(\frac{1}{2}\|\cdot\|_{\mathrm{F}}^2 \square\, \Psi\right)(Y) = Y - \mathrm{prox}_{\Psi}(Y).$$

According to Moreau decomposition (Moreau, 1962), it moreover holds that for all $Y \in \mathbb{R}^{n\times s}$,

$$Y - \mathrm{prox}_{\Psi}(Y) = \mathrm{prox}_{\Psi^\star}(Y).$$

Thus the DCA algorithm can be applied as long as the computation of $\mathrm{prox}_{\Psi^\star}$ is possible. The correspondence between the choice of objective and the resulting proximal operator is gathered in Table 1. In particular, one can note that tractability of the DCA algorithm requires that the projection on the balls for the dual norm $\|\cdot\|_\star$ is possible in the Huber case, whereas for ϵ-insensitive objectives what matters is being able to project on the balls for the norm $\|\cdot\|$. This requirement drives the choice of the norm used in the definition of the objectives, as projecting on balls can be obtained in closed-form when the norm is a 2-norm or an ∞-norm. The DCA is summarized in Algorithm 1.

5. Numerical Experiments

Through numerical evaluations, we show the efficiency and flexibility of the proposed KPCA dualization on a diverse collection of dimensionality reduction tasks.

First, we evaluate the efficiency of the resulting LBFGS-based optimization algorithm for KPCA on multiple datasets, with comparisons to other standard KPCA solvers. We apply the proposed algorithm to problems of different sizes, and we also study the effect on performance of the decay of the eigenspectrum of G.

Then, we quantify the robustness of the Huber losses on data contaminated with outliers depending on the noise level, and we study the influence of the loss parameters. Regarding the ϵ-insensitive losses, we study the accuracy-sparsity tradeoff w.r.t. the ϵ parameter and the number of components s.

Experiments are implemented in Python 3.10 on a machine with a 3.7GHz Intel i7-8700K processor and 64GB RAM. The code is available at `https://github.com/taralloc/dc-kpca`.

Datasets. We evaluate our approach on synthetic and real-world datasets. Synth 1 ($n = 7000, d = 10000$) is a high-dimensional synthetic dataset where samples are drawn randomly from a multivariate normal distribution with zero mean and fixed covariance matrix. For real-world data, we consider datasets from LIBSVM (Chang & Lin, 2011), UCI (Dua & Graff, 2017), and common deep learning benchmarks: Iris ($n = 150, d = 4$), the bioinformatics dataset Protein ($n = 14895, d = 357$), the text categorization dataset RCV1 ($n = 20242, d = 47236$), and the outputs of the second to last layer of a ResNet18 (He et al., 2016) trained on the computer vision dataset CIFAR-10 ($n = 60000, d = 512$).

Experimental setups. Regarding optimization for KPCA with the square loss, we employ the LBFGS algorithm with backtracking linesearch using the strong Wolfe conditions with initialization from the standard normal distribution. For the Moreau envelopes, i.e. Huber and ϵ-insensitive losses, we employ the DC algorithm; for faster convergence second order optimization algorithms for this problem with composite smooth + non-smooth structure can also be employed, e.g. (Stella et al., 2017). For the square loss, the accuracy of a dual iterate H_k from our solver at each iteration k is chosen as the relative difference $\eta = |d(H_k) - d_{\mathrm{opt}}|/d_{\mathrm{opt}}$ between the dual cost $d(H_k) = \frac{1}{2}\,\mathrm{Tr}\, H_k^\top H_k - \mathrm{Tr}\sqrt{H_k^\top G H_k}$ at iteration k and the optimal dual cost $d_{\mathrm{opt}} = -\frac{1}{2}\sum_{i=1}^s \lambda_i$, with λ_i being the i-th largest eigenvalue of G. In fact, the optimal primal cost is the highest variance in s components, i.e., $\frac{1}{2}\sum_{i=1}^s \lambda_i$, and strong duality holds in our dualization. For all used solvers, we use the same stopping criterion based on achieving a target tolerance. For the other convoluted losses, we stop the DCA when the absolute variation of the loss is less than machine precision with at most 1000 iterations.

Table 2: **KPCA Training Time**. Runtime for multiple KPCA problems with higher tolerance. Speedup factor w.r.t. RSVD.

Task	n	Time (s) for $\delta = 10^{-2}$				Speedup
		SVD	Lanczos	RSVD	Ours	Factor
Synth 1	7000	96.73	0.85	1.97	**0.53**	3.72
Protein	14895	868.64	3.46	6.70	**1.07**	6.25
RCV1	20242	-	6.04	12.50	**2.12**	5.90
CIFAR-10	60000	-	48.10	123.89	**13.51**	9.17

Table 3: **KPCA Training Time**. Runtime for multiple KPCA problems with lower tolerance.

Task	n	Time (s) for $\delta = 10^{-4}$				Speedup
		SVD	Lanczos	RSVD	Ours	Factor
Synth 1	7000	96.73	4.78	1.97	**0.56**	3.52
Protein	14895	868.64	3.51	6.95	**1.07**	6.59
RCV1	20242	-	19.72	12.75	**3.78**	3.38
CIFAR-10	60000	-	48.15	122.58	**29.26**	4.19

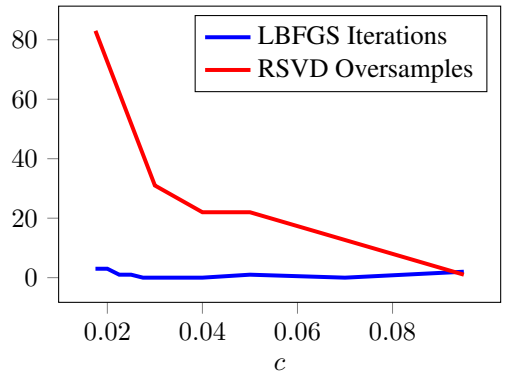

Figure 1: **Varying eigenspectrum.** Additional computational burden when the spectrum of G changes (larger c corresponds to spectra with faster decay). Blue: our LBGS-based algorithm, red: randomized SVD.

5.1. More Efficient KPCA

This subsection demonstrates the performance of the proposed LBFGS-based algorithm to solve the KPCA problem (1). We compare our method with three common KPCA solvers: full SVD (SVD), Implicitly Restarted Lanczos Method (Lehoucq et al., 1998), and randomized SVD (RSVD) (Halko et al., 2011). The Lanczos solver finds the first s eigenvalues and corresponding eigenvectors of the symmetric matrix G, while randomized SVD finds the truncated singular value decomposition of G by random projections. These solvers rely on different stopping criteria. To make a fair comparison, note that $d_{\text{opt}} = d(U\sqrt{S})$, with $G = USV^\top$ being the SVD of G. Therefore, the accuracy of an approximate solution $\hat{U}, \hat{S}, \hat{V}$ is measured by the following relative dual cost residual:

$$\eta = \left| d(\hat{U}\sqrt{\hat{S}}) - d_{\text{opt}} \right| / d_{\text{opt}}, \tag{10}$$

where for RSVD $\hat{S}$ is the diagonal matrix of the largest s singular values of G and corresponding computed singular vectors $\hat{U}, \hat{V}$, while for the eigendecomposition solver $\hat{S}$ is the diagonal matrix of the largest s eigenvalues of G and corresponding eigenvectors $\hat{U} = \hat{V}$ found by the Lanczos solver. Full SVD is run to machine precision for comparison. For a given tolerance δ, we stop training when $\eta < \delta$. In particular, for RSVD, the number of required oversamples is found by increasing the number of oversamples until the target tolerance is reached.

The kernel is chosen to be the Laplace kernel $k(z, y) = \exp\left(-\|z - y\|_2 / (2\sigma^2)\right)$ with $\sigma = 0.1\sqrt{d\sigma_x}$ and σ_x the variance of the training data. For the KPCA dimensionality reduction task, it is common to assume that the given high-dimensional data can be expressed over a small number of principal components; therefore, in these experiments, we use $s = 20$. The experiments are averaged over 5 runs.

Table 2 and 3 show the training times on different KPCA tasks for multiple tolerance levels $\delta = 10^{-2}, 10^{-4}$; lowest training times are in bold. The time to compute the full SVD to machine precision is given for reference in the SVD column, where "-" indicates that SVD took longer than 30 minutes. The speedup factor is $t^{(\text{RSVD})}/t^{(\text{LBFGS})}$, where $t^{(\text{RSVD})}, t^{(\text{LBFGS})}$ is the training time using the RSVD solver and our LBFGS-based solver, respectively. For tolerance $\delta = 10^{-2}$, our solver is faster than all other KPCA solvers and at least 3 times faster than RSVD. For the lowest tolerance $\delta = 10^{-4}$, our solver is the fastest with generally

smaller speedup factor. Since our solver is second order, it can reach higher accuracy quickly, with subsequent iterations giving relatively lower improvement, which explains why the speedup is more apparent at higher tolerances. All solvers can achieve high accuracy; however, in some tasks (e.g., CIFAR-10 and Protein) the Lanczos method cannot take significant advantage of higher tolerance requirements.

A solver's performance depends on the properties of the given data and kernel. For instance, it is known that RSVD requires more oversamples for matrices whose eigenspectrum decays slowly (Halko et al., 2011), which is often the case in real-world problems. On the other hand, our LBGS-based solver does not suffer from such issue. To further illustrate this point, we construct G from random data X as $G = 0.01(X + X^\top) + UDU^\top$, where U is any orthogonal matrix and D is a diagonal matrix s.t. $D_{ii} = \exp(-ci)$. Varying c controls how quickly the spectrum of G decays, with lower c corresponding to slower decays. In Fig. 1, we vary c and show the additional computational cost for RSVD and for our solver when the spectrum of G changes. For RSVD, the red line shows the additional required oversamples needed to achieve tolerance $\delta = 10^{-4}$ w.r.t. the experiment with lowest number of oversamples. The blue line shows the additional LBFGS iterations required by our solver to achieve the same tolerance. While RSVD requires significantly more oversamples to reach a fixed accuracy as c decreases, our solver takes a similar number of iterations, showing that our solver is mostly unaffected by this type of change in the eigenspectrum of G.

The influence of s on training time is studied in Figure 2 on random data with $n = 15000$: higher s leads to longer training times for both LBFGS and RSVD, with LBFGS maintaining performance advantage. Very high s impacts LBFGS's training time as it needs to compute the SVD of a $s \times s$ matrix at every iteration.

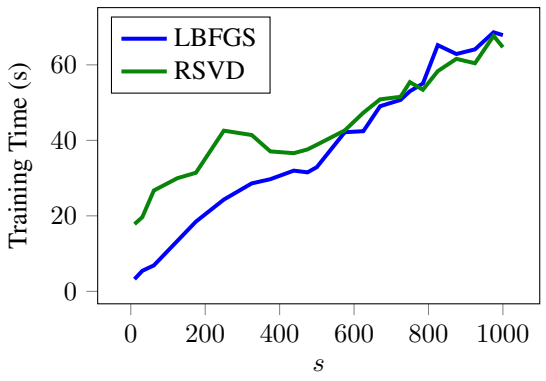

Figure 2: **KPCA Training Time**. Effect of s.

Table 4: **Robustness**. MSE on contaminated Iris dataset.

τ	$\frac{1}{2}\|\cdot\|^2$	H^2_κ	H^1_κ
10	7.591059	6.833484	7.381284
25	7.910846	7.182663	7.687518
50	8.691805	8.045477	8.430957
75	9.782740	9.259353	9.465706
100	11.183650	10.824293	10.791766

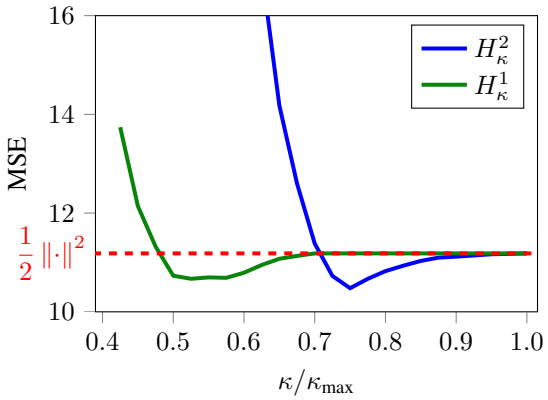

Figure 3: **Robustness**. Effect of κ for the loss H^2_κ and H^1_κ.

5.2. Huber Losses

We now present numerical experiments to illustrate the flexibility of the KPCA dualization framework applied to the Huber loss. We consider the Huber losses associated to different norms: H^1_κ is the Moreau envelope of $H \mapsto \kappa \|H\|_1$ and H^2_κ is the Moreau envelope of $H \mapsto \max_{i\in[n]} \|h_i\|_2$. Both cases are optimized with the same DC algorithm.

We first investigate the robustness induced by the Huber losses on the Iris dataset. We contaminate the data in the following way: to corrupt data $\{x_i\}_{i=1}^n$, we first draw a set $I \subset \{0, ..., n\}$ s.t. $|I| = \lfloor \omega n \rfloor$, with $\omega \in [0, 1]$ being the proportion of corrupted samples. Then, we introduce outliers using multiplicative Gaussian noise with zero mean and τ standard deviation: for $i \in I$, x_i is replaced by $b_i x_i$, where b_i is drawn from $\mathcal{N}(0, \tau^2)$. We evaluate performance by reconstruction error (MSE) in input space of non-contaminated test samples: higher MSE means that the learned subspace is influenced by the outliers, while lower MSE corresponds to more robust models. We compute pre-images using the technique of (Bakır et al., 2004). As discussed in Section 3, as κ grows, the constraint on the dual variables becomes void and we recover the square loss KPCA problem. Therefore, we set $\kappa = 0.6\kappa_{\max}$, $0.8\kappa_{\max}$ for H^1_κ, H^2_κ, respectively.

The resulting test MSE values are given in Table 4 for $\omega = 8\%$ on a 20% test split. We employ a Gaussian kernel

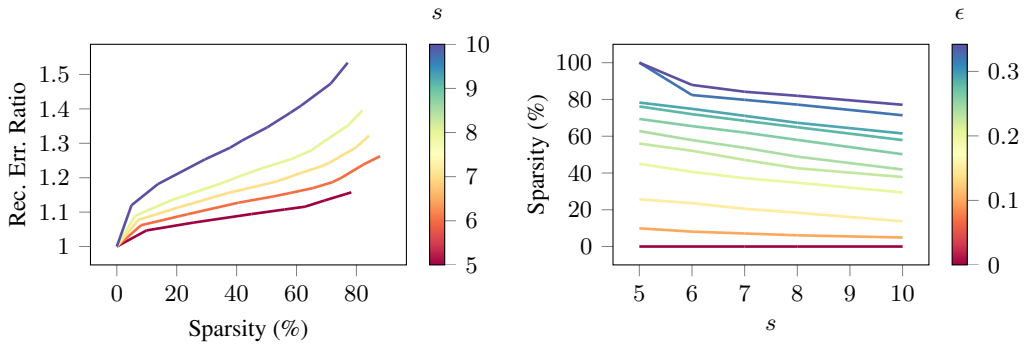

Figure 4: **Sparsity**. Reconstruction error for the ℓ_ϵ^2 loss for multiple ϵ and s.

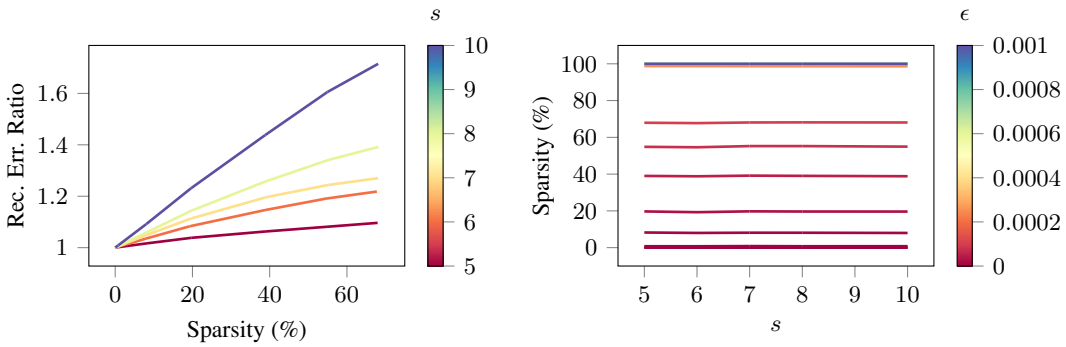

Figure 5: **Sparsity**. Reconstruction error for the ℓ_ϵ^∞ loss for multiple ϵ and s.

$k(z, y) = \exp\left(-\|z - y\|_2^2/2\right)$ for all models. The Huber losses H_κ^1, H_κ^2 are more robust to outliers for all noise levels. Particularly, H_κ^1 show greater robustness at the highest noise level, while lower noise levels favor H_κ^2. Overall, these results show that the learned principal components are more robust to the introduced outliers.

The influence of κ is studied in Fig. 3 with $\tau = 100$. The Huber losses (blue and green lines) can be distinctively more robust to outliers than the square loss (red dashed line). When κ becomes closer to $\kappa_{\max}$, the robustness effect is void and the MSE converges to the one from the standard KPCA. With small κ, the constraint on the dual variables is too strict to learn meaningful principal components, as the size of the projection ball becomes insufficient. Accordingly, the MSE grows quickly in the small κ region. A balanced choice of κ results in lower MSE, i.e., reduces the influence of the outliers and therefore learns more robust principal components.

5.3. ϵ-insensitive Losses

The ϵ-insensitive losses induce sparsity on H. The choice of the ball used to define the Moreau envelope determines the sparsity type through the dual norm and ϵ affects the sparsity level, where $\epsilon = 0$ recovers the square loss case. In particular, for the dual norms $\|H\|_\star = \|H\|_1$ and $\|H\|_\star = \sum_{i=1}^n \|h_i\|_2$, we obtain respectively the losses ℓ_ϵ^∞, ℓ_ϵ^2. The former promotes unstructured sparsity as the associated proximal step involves coordinate-wise soft-thresholding, the latter promotes block sparsity as the associated proximal step involves block soft-thresholding. The use of convoluted losses makes it possible to cast both the robustness and sparsity losses in the same duality framework. Accordingly, as for the Huber loss family, we address this optimization problem through the DC algorithm.

The study of the role of ϵ and s for the ℓ_ϵ^2 loss is shown in Fig. 4, where we consider a random data matrix X with $n = 1000$, $d = 20$ and the kernel is chosen to be Gaussian. The reconstruction error ratio is the ratio between the reconstruction error using the components learned with the ℓ_ϵ^2 loss and the reconstruction error from the square loss. Here, sparsity is in terms of percentage of zero rows in H. In fact, typically the dual solution from KPCA is dense and all training points, corresponding to the rows of H, contribute. Block sparsity therefore induces a representation in a fraction of the training points. We can see that setting $\epsilon = 0$ recovers the square loss case. As ϵ grows, H becomes sparser. We also expect that, as ϵ grows, the performance declines in terms of reconstruction error, demonstrating a trade-off between sparsity and accuracy of the learned repre-

sentation. Such compromise depends on the requirements of specific applications. In the example of Fig. 4, with $s = 5$ principal components, one can obtain 40% sparsity with just a 10% increase in reconstruction error. Increasing s with fixed ϵ usually leads to a slight decrease in sparsity, meaning that a higher ϵ is needed to achieve a similar sparsity level. The ℓ_ϵ^2 performance relative to the square loss declines with higher s for a fixed sparsity level. Similar conclusions can be drawn for the ℓ_ϵ^∞ loss from Fig. 5, where sparsity is in terms of zero entries of H.

6. Conclusion

This work presents a duality framework for the kernel PCA problem seen as a difference of convex functions. The generalized family of objectives with Moreau envelopes structure allows to extend the variance maximization problem to a wider choice of objectives, inducing robust and sparse estimators. The resulting gradient-based algorithm for standard KPCA shows important speedups in training time compared to the SVD solvers. Future work could focus on convergence properties of the DC algorithm towards optimal critical points.

Acknowledgements

This work is jointly supported by ERC Advanced Grant E-DUALITY (787960), iBOF project Tensor Tools for Taming the Curse (3E221427), Research Council KU Leuven: Optimization framework for deep kernel machines C14/18/068, KU Leuven Grant CoE PFV/10/002, and Grant FWO G0A4917N, EU H2020 ICT-48 Network TAILOR (Foundations of Trustworthy AI - Integrating Reasoning, Learning and Optimization), and the Flemish Government (AI Research Program), and Leuven.AI Institute. This work was also supported by the Research Foundation Flanders (FWO) research projects G086518N, G086318N, and G0A0920N; Fonds de la Recherche Scientifique — FNRS and the Fonds Wetenschappelijk Onderzoek — Vlaanderen under EOS Project No. 30468160 (SeLMA).

References

Alzate, C. and Suykens, J. A. Kernel component analysis using an epsilon-insensitive robust loss function. *IEEE Transactions on Neural Networks*, 19(9):1583–1598, 2008.

Bakır, G. H., Weston, J., and Schölkopf, B. Learning to find pre-images. *Advances in Neural Information Processing Systems*, 16:449–456, 2004.

Beck, A. and Teboulle, M. Dual randomized coordinate descent method for solving a class of nonconvex problems. *SIAM Journal on Optimization*, 31(3):1877–1896, 2021.

Chang, C.-C. and Lin, C.-J. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):1–27, 2011.

Chin, T.-J. and Suter, D. Incremental kernel principal component analysis. *IEEE Transactions on Image Processing*, 16(6):1662–1674, 2007.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Fan, J. and Chow, T. W. S. Exactly Robust Kernel Principal Component Analysis. *IEEE Transactions on Neural Networks and Learning Systems*, 31(3):749–761, 2020. ISSN 2162-2388.

Gemp, I., McWilliams, B., Vernade, C., and Graepel, T. Eigengame: PCA as a nash equilibrium. In *International Conference on Learning Representations*, 2021.

Günter, S., Schraudolph, N. N., and Vishwanathan, S. V. N. Fast iterative kernel principal component analysis. *Journal of Machine Learning Research*, 8(66):1893–1918, 2007.

Guo, L., Wu, P., Gao, J., and Lou, S. Sparse kernel principal component analysis via sequential approach for nonlinear process monitoring. *IEEE Access*, 7:47550–47563, 2019a.

Guo, L., Wu, P., Gao, J., and Lou, S. Sparse kernel principal component analysis via sequential approach for nonlinear process monitoring. *IEEE Access*, 7:47550–47563, 2019b.

Halko, N., Martinsson, P. G., and Tropp, J. A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Honeine, P. Online kernel principal component analysis: A reduced-order model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1814–1826, 2012.

Kim, C. and Klabjan, D. A simple and fast algorithm for L1-Norm Kernel PCA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(8):1842–1855, 2020.

Laforgue, P., Lambert, A., Brogat-Motte, L., and d'Alché Buc, F. Duality in RKHSs with infinite dimensional outputs: Application to robust losses. In *International Conference on Machine Learning (ICML)*, pp. 5598–5607, 2020.

Lambert, A., Bouche, D., Szabo, Z., and D'Alché-Buc, F. Functional output regression with infimal convolution: Exploring the huber and epsilon-insensitive losses. In *International Conference on Machine Learning*, 2022.

Langone, R. and Suykens, J. A. K. Fast kernel spectral clustering. *Neurocomputing*, 268:27–33, December 2017.

Lehoucq, R. B., Sorensen, D. C., and Yang, C. *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*. SIAM, 1998.

Lewis, A. S. Derivatives of spectral functions. *Mathematics of Operations Research*, 21(3):576–588, 1996.

Moreau, J. J. Fonctions convexes duales et points proximaux dans un espace hilbertien. *Comptes rendus hebdomadaires des séances de l'Académie des sciences*, 255: 2897–2899, 1962.

Moreau, J.-J. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société mathématique de France*, 93:273–299, 1965.

Nguyen, M. and Torre, F. Robust kernel principal component analysis. *Advances in Neural Information Processing Systems*, 21, 2008.

Sangnier, M., Fercoq, O., and d'Alché-Buc, F. Data sparse nonparametric regression with ϵ-insensitive losses. In *Asian Conference on Machine Learning (ACML)*, pp. 192–207, 2017.

Schölkopf, B., Smola, A., and Müller, K.-R. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10(5):1299–1319, July 1998. ISSN 0899-7667.

Smola, A. J., Mangasarian, O. L., and Schölkopf, B. Sparse kernel feature analysis. In Gaul, W. and Ritter, G. (eds.), *Classification, Automation, and New Media*, pp. 167–178, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

Stella, L., Themelis, A., Sopasakis, P., and Patrinos, P. A simple and efficient algorithm for nonlinear model predictive control. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 1939–1944, 2017.

Tao, P. D. and An, L. T. H. Convex analysis approach to DC programming: Theory, algorithms and applications. *Acta mathematica vietnamica*, 22(1):289–355, 1997.

Thiao, M., Dinh, T. P., and Thi, H. A. L. A DC programming approach for sparse eigenvalue problem. In *International Conference on International Conference on Machine Learning*, 2010.

Tipping, M. Sparse kernel principal component analysis. *Advances in Neural Information Processing Systems*, 13, 2000.

Toland, J. F. On subdifferential calculus and duality in nonconvex optimization. *Bull. Soc. Math. France, Mémoire*, 60:177–183, 1979.

Uschmajew, A. Well-posedness of convex maximization problems on stiefel manifolds and orthogonal tensor product approximations. *Numerische Mathematik*, 115(2): 309–331, 2010.

Wang, D. and Tanaka, T. Sparse kernel principal component analysis based on elastic net regularization. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 3703–3708, 2016.

Wang, D. and Tanaka, T. A robust method for kernel principal component analysis. *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pp. 294–297, 2020a.

Wang, D. and Tanaka, T. Robust kernel principal component analysis with $\ell_{2,1}$-regularized loss minimization. *IEEE Access*, 8:81864–81875, 2020b.

Wright, J. and Ma, Y. *High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications*. Cambridge University Press, 2022.

A. Proofs

A.1. Proof of Proposition 3.1

Proof. Since f is proper closed convex, it holds that $f = f^{**}$, so equivalently we can write Problem 2 as

$$\inf_{W\in\mathcal{U}} g(W) - \sup_{V\in\mathcal{K}} \{\langle V, \Gamma W\rangle - f^*(V)\} = \inf_{W\in\mathcal{U}} g(W) + \inf_{V\in\mathcal{K}} \{f^*(V) - \langle V, \Gamma W\rangle\} \tag{11}$$

$$= \inf_{W\in\mathcal{U}, V\in\mathcal{K}} g(W) + f^*(V) - \langle V, \Gamma W\rangle \tag{12}$$

$$= \inf_{V\in\mathcal{K}} f^*(V) - \sup_{W\in\mathcal{U}} \{\langle \Gamma^\sharp V, W\rangle - g(W)\} \tag{13}$$

$$= \inf_{V\in\mathcal{K}} f^*(V) - g^*(\Gamma^\sharp V). \tag{14}$$

□

A.2. Proof of Proposition 3.2

We begin by expliciting a formula for $\Gamma^\sharp$.

Lemma A.1. *Let $H \in \mathbb{R}^{n\times s}$. Then*

$$\Gamma^\sharp H = \left[\sum_{i=1}^n h_{ij}\phi(x_i)\right]_{j=1}^s \in \mathcal{H}^s.$$

Proof. Let $W = [w_j]_{j=1}^s \in \mathcal{H}^s$ and $H \in \mathbb{R}^{n\times s}$. It holds that

$$\begin{aligned}
\langle H, \Gamma W\rangle_{\mathbb{R}^{n\times s}} &= \sum_{i=1}^n \sum_{j=1}^s h_{ij}[\Gamma W]_{ij} \\
&= \sum_{i=1}^n \sum_{j=1}^s h_{ij}\langle \phi(x_i), w_j\rangle_{\mathcal{H}} \\
&= \sum_{j=1}^s \left\langle \sum_{i=1}^n h_{ij}\phi(x_i), w_j \right\rangle_{\mathcal{H}} \\
&= \left\langle \left[\sum_{i=1}^n h_{ij}\phi(x_i)\right]_{j=1}^s, W \right\rangle_{\mathcal{H}^s}
\end{aligned}$$

which conludes the proof. □

We are now ready to prove that $g^\star(\Gamma^\sharp H) = \operatorname{Tr}\left(\sqrt{H^\top G H}\right)$.

Proof. Let $W = [w_j]_{j=1}^s \in \mathcal{H}^s$ and $H \in \mathbb{R}^{n\times s}$.

$$\sup_{\mathcal{G}(W)\preccurlyeq I_s} \langle \Gamma^\sharp H, W\rangle_{\mathcal{H}^s} = \sup_{\mathcal{G}(W)\preccurlyeq I_s} \sum_{j=1}^s \langle w_j, \sum_{i=1}^n h_{ij}\phi(x_i)\rangle_{\mathcal{H}}$$

Since $\mathcal{H} = \operatorname{Span}\left(\{\phi(x_i)\}_{i=1}^n\right) \bigoplus \left(\operatorname{Span}\left(\{\phi(x_i)\}_{i=1}^n\right)\right)^\top$ and using orthogonality properties, we can parameterize W using a matrix $A \in \mathbb{R}^{n\times s}$ so that

$$\forall j \in [s], w_j = \sum_{i=1}^n a_{ij}\phi(x_i).$$

Then

$$\begin{aligned}
\langle \Gamma^\sharp H, W\rangle_{\mathcal{H}^s} &= \sum_{j=1}^s \langle \sum_{i=1}^n h_{ij}\phi(x_i), \sum_{l=1}^n a_{lj}\phi(x_l)\rangle_{\mathcal{H}} \\
&= \operatorname{Tr} A^\top G H.
\end{aligned}$$

Moreover, $\mathcal{G}(W) = A^\top GA$. We thus fall back on the problem

$$\sup_{A^\top GA \preccurlyeq I_s} \operatorname{Tr} A^\top GH.$$

By performing the change of variable $B = G^{\frac{1}{2}} A$ (possible since G is full rank), the problem becomes

$$\sup_{B^\top B \preccurlyeq I_s} \operatorname{Tr} B^\top G^{\frac{1}{2}} H.$$

We recognize the Fenchel-Legendre transform of the indicator of the convex hull of the unit ball for the spectral norm, which is known to be the Schatten 1-norm (also known as nuclear norm) that we denote $\|\cdot\|_{S_1}$ in what follows. We have thus proven that

$$g^\star(\Gamma^\sharp H) = \left\| G^{\frac{1}{2}} H \right\|_{S_1}.$$

A simple application of the SVD decomposition then suffices to see that

$$\left\| G^{\frac{1}{2}} H \right\|_{S_1} = \operatorname{Tr} \sqrt{H^\top GH},$$

which conludes. □

A.3. Proof of Proposition 3.4

Proof. We begin by rewriting $\pi(H) = R \circ c(H)$ where $R(X) = \operatorname{Tr} \sqrt{X}$ and $c(H) = H^\top GH$. Note that $R(X) = \operatorname{Tr} \sqrt{X} = \sum_{i=1}^s \sqrt{\lambda_i(X)}$, with $\lambda_i(x)$ the i-th largest eigenvalue of X. Also, $R(X) = r(\lambda(X))$, with $r(z) = \sum_{i=1}^s \sqrt{z_i}$, where z_i indicates the i-th component of z and $\lambda(X)$ is the eigenvalues function defined as $\lambda(X) = [\lambda_1(X), \lambda_2(X), \dots, \lambda_s(X)]^\top$. Note that r is symmetric to permutations, so R is a symmetric spectral function over the set of symmetric matrices $\mathbb{S}^s$. Therefore, the gradient of R is (see Theorem 1.1 in (Lewis, 1996)):

$$\nabla R(X) = U^\top \operatorname{diag}(\nabla r(\lambda(X)))U, \tag{15}$$

with U any orthogonal matrix such that $X = U^\top \operatorname{diag}(\lambda(X))U$. The gradient of r is

$$\nabla r(z) = \frac{1}{2\sqrt{z}}, \tag{16}$$

for $z > 0$ (which is the case for G positive definite and H full rank). Moreover

$$\nabla c(H) = 2GH. \tag{17}$$

Finally by the chain rule we have

$$\nabla \pi(H) = 2GHU^\top \operatorname{diag}\left(\frac{1}{2\sqrt{\lambda(H^\top GH)}}\right)U, \tag{18}$$

where $U \in \mathbb{R}^{s \times s}$ is any orthogonal matrix satisfying

$$H^\top GH = U^\top \operatorname{diag}\left(\lambda(H^\top GH)\right) U. \tag{19}$$

Therefore,

$$\nabla \pi(H) = GHU^\top DU, \tag{20}$$

with $D = \operatorname{diag}\left(\frac{1}{\sqrt{\lambda(H^\top GH)}}\right)$. □

A.4. Proof of Proposition 3.7

Proof. Writing the dual cost with $H = U_r^\top \sqrt{\Sigma_r}$, we get

$$\begin{aligned}\frac{1}{2}\operatorname{Tr}(H^\top H) - \operatorname{Tr}(\sqrt{(H^\top G H)}) &= \frac{1}{2}\operatorname{Tr}(\sqrt{\Sigma_r}U_r U_r^\top \sqrt{\Sigma_r}) - \operatorname{Tr}(\sqrt{\sqrt{\Sigma_r}U_r G U_r^\top \sqrt{\Sigma_r}}) \\ &= \frac{1}{2}\operatorname{Tr}(\Sigma_r) - \operatorname{Tr}\sqrt{\Sigma_r^2} \\ &= -\frac{1}{2}\operatorname{Tr}(\Sigma_r).\end{aligned}$$

As this quantity correspond to (minus) the primal cost and srong duality holds, then H is a dual solution. □

A.5. Proof of Proposition 3.6

Proof. First remark that for all $H \in \mathbb{R}^{n\times s}$,

$$\nabla J(H) = HH^\top H - GH.$$

Let $\hat{H}$ be a critical point of the dual function, we denote by $\hat{H}^\top G\hat{H} = U^\top S U$ the SVD of $\hat{H}^\top G\hat{H}$. The condition $0 \in \partial\left(H \mapsto \frac{1}{2}\|H\|_{\mathrm{F}}^2 - \pi(H)\right)(\hat{H})$ implies that

$$\hat{H} = G\hat{H}U^\top S^{-\frac{1}{2}}U$$

Thus

$$\begin{aligned}\hat{H}^\top\hat{H} &= \hat{H}^\top G\hat{H}U^\top S^{-\frac{1}{2}}U \\ &= U^\top S U U^\top S^{-\frac{1}{2}}U \\ &= U^\top S^{\frac{1}{2}}U,\end{aligned}$$

so that

$$\hat{H}\hat{H}^\top\hat{H} = G\hat{H},$$

showing that $\nabla J(\hat{H}) = 0$, which concludes the proof. □

A.6. Proof of Proposition 3.8

Proof. Let $\hat{H}$ be a solution to the dual problem. We know that the primal solution satisfies

$$\hat{W} \in \underset{W\in\mathcal{H}^s, \mathcal{G}(W)=I_s}{\arg\max} \langle W, \Gamma^\sharp\hat{H}\rangle.$$

We can express each $\hat{w}_j = \sum_{i=1}^n \hat{a}_{ij}\phi(x_i)$, and the problem becomes

$$\max_{A\in\mathbb{R}^{n\times s}} \langle A, G\hat{H}\rangle \quad \text{s.t. } A^\top G A = I_s.$$

By doing the change of variables $B = G^{\frac{1}{2}}A$, we want to solve

$$\max_{B\in\mathbb{R}^{n\times s}} \langle B, G^{\frac{1}{2}}\hat{H}\rangle \quad \text{s.t. } B^\top B = I_s \tag{21}$$

Let $G^{\frac{1}{2}}\hat{H} = V^\top S U$ be the SVD decomposition of $G^{\frac{1}{2}}\hat{H}$. We have that $\hat{B} = V^\top U$ maximizes 21, which in turn gives $\hat{A} = G^{-\frac{1}{2}}V^\top U$.
We can now express $\hat{A}$ using $\hat{H}$, as

$$\hat{H} = G^{-\frac{1}{2}}V^\top S U$$

implies that

$$\hat{H}U^\top S^{-1}U = \hat{A}.$$

Remarking that $\hat{H}^\top G\hat{H} = U^\top S^2 U$ is the SVD of $\hat{H}^\top G\hat{H}$ allows to conclude the proof. □

B. More Experimental Results

B.1. Kernel Choice

Numerical evaluations are conducted to assess the effect of different kernel functions. In fact, different kernels give different eigenspectra of the Gram matrix. In general, numerical algebra solvers' accuracy and required number of iterations may depend on the eigenspectrum of G. For example, Randomized SVD (RSVD) is known to be less accurate with data matrices whose eigenspectrum decays slowly (Halko et al., 2011), which is often the case with real-world noisy data. On the other hand, in the main body we show that our method is mostly unaffected by the shape of the eigenspectrum of G.

We conduct further experiments to verify that our method is effective when we use other kernels as well. The results in Table 5 below show the KPCA training time with Gaussian kernel, while in Table 2 in the main body the kernel was set as the Laplacian. The speedups are clear also with the Gaussian kernel.

Table 5: KPCA Training Time with Gaussian kernel for $\delta = 10^{-2}$. Speedup factor w.r.t. RSVD.

Task	n	Time (s)				Speedup
		SVD	Lanczos	RSVD	Ours	Factor
Synth 1	7000	87.33	1.05	3.04	**0.56**	5.41
Protein	14895	979.84	3.86	10.61	**1.08**	9.81
RCV1	20242	-	21.03	19.69	**3.76**	5.23
CIFAR-10	60000	-	47.94	197.99	**13.47**	14.70

Moreover, the results in Table 6 below show the test MSE on the corrupted Iris dataset used in Section 5.2 in the main body with Laplacian kernel, while in the main body was set as the Gaussian. The Huber loss shows improved robustness also with the Laplacian kernel.

Table 6: MSE on contaminated Iris dataset using Huber loss with Laplacian kernel.

τ	$\frac{1}{2}\|\cdot\|^2$	H^2_κ	H^1_κ
10	3.024668	2.862165	2.762514
25	4.580805	4.415508	4.337033
50	8.998080	8.824152	8.808865
75	15.695008	15.507481	15.590248
100	24.671558	24.671039	24.671153

B.2. Comparative Analysis

We conduct comparative experiments between our sparse/robust KPCA with recent sparse/robust KPCA methods. First, we compare with $\ell_{2,1}$-RKPCA (Wang & Tanaka, 2020b) and with L1-KPCA (Kim & Klabjan, 2020), which are recent relevant robust KPCA methods from the literature that have been shown experimentally to exhibit good robustness for this task. The hyperparameter in (Wang & Tanaka, 2020b), which controls the importance of the $\ell_{2,1}$-norm penalization term, is selected in the range suggested by the authors in their paper. In Table 7, we report reconstruction error (MSE) for the Iris data, contaminated as described in Section 5.2 in the main body for multiple noise levels. Higher MSE means that the learned subspace is influenced by the outliers, while lower MSE corresponds to more robust models.

Additionally, we compare with SSKPCA (Guo et al., 2019a) and with SKPCA (Wang & Tanaka, 2016), which are recent relevant sparse KPCA methods from the literature. In Table 8, the reconstruction error ratio is the ratio between the reconstruction error using the components learned with the corresponding sparse KPCA method and the reconstruction error from the dense problem. Ratios closer to 1 are better. The settings and dataset are the ones from Section 5.3 in the main body. The sparsity parameter λ in SKPCA and γ in SSKPCA are selected to obtain a fixed sparsity percentage in the matrix of coefficients.

The robust KPCA in (Kim & Klabjan, 2020) uses L1-norm KPCA, i.e., maximizing variance with respect to the L1-norm, while we consider variance-like objectives with infimal convolution with the squared norm, so we can recover the standard

Table 7: Robust KPCA comparison. MSE on contaminated Iris dataset.

τ	$\ell_{2,1}$-RKPCA	L1-KPCA	$\frac{1}{2}\|\cdot\|^2$	H^2_κ	H^1_κ
10	6.965341	6.829537	7.591059	6.833484	7.381284
25	7.437400	7.598856	7.910846	7.182663	7.687518
50	8.113397	8.085395	8.691805	8.045477	8.430957
75	9.734140	9.547392	9.782740	9.259353	9.465706
100	10.934794	10.855314	11.183650	10.824293	10.791766

Table 8: Sparse KPCA comparison. Reconstruction error ratio.

Sparsity (%)	SKPCA	SSKPCA	ℓ^2_ϵ	ℓ^∞_ϵ
10	1.19053	1.17019	1.04675	1.01685
20	1.67440	1.18649	1.06966	1.03824
30	1.83307	1.18794	1.08153	1.04247
40	2.30358	1.20480	1.09440	1.06308
50	2.75248	1.22205	1.10164	1.08127

KPCA as a special case. The method in (Wang & Tanaka, 2020b) introduces extra variables and multiple $\ell_{2,1}$-norm penalizations in the objective and consequently multiple penalty hyperparameters which makes it harder to tune than our proposed method with a single parameter κ; the optimization is performed with alternating updates with three additional optimization subproblems, which require to compute the SVD of an $n \times s$ matrix at each iteration. SKPCA (Wang & Tanaka, 2016) and SSKPCA (Guo et al., 2019a) relax and modify KPCA to use ElasticNet optimization for promoting sparsity. To get sparsity, SKPCA make use of the matrix $G^{-1/2}$, which requires computing the full SVD of G, and SSKPCA's iterative algorithm is initialized with the top s eigenvectors of G, requiring to compute the truncated SVD of G. In comparison, we get sparsity without performing such costly operation, as our modelling considers the SVD of the much smaller $s \times s$ matrix $H^\top GH$. Overall, our formulation performs similarly without having to compute the full SVD.