Report: Extending Kernel PCA through Dualization

CELANIE Erwan[1], **NEL Louis**[1]

[1]Institut Polytechnique de Paris – FR – France

erwan.celanie@polytechnique.edu, louis.nel@polytechnique.edu

Details of the original paper:

- *Title:* Extending Kernel PCA through Dualization: Sparsity, Robustness and Fast Algorithms
- *Authors:* F. Tonin, A. Lambert, P. Patrinos, J. A. K. Suykens
- *Publication venue:* Proceedings of the 40^{th} International Conference on Machine Learning

1. Introduction

Kernel PCA is an extension of the standard Principal Components Analysis technique that takes advantage of kernel methods. It makes use of an implicit feature mapping into a Hilbert space (potentially infinite-dimensional), making PCA more flexible while keeping the computations feasible, thanks to the kernel trick.

In this report, we study the paper [Ton+23], which places KPCA within the framework of Difference of Convex function (DC) algorithms. This presents two advantages: first, the computations are faster than the usual SVD method for doing KPCA. Second, desirable properties properties such as sparsity and robustness are easily incorporated into this framework.

2. Background on KPCA

Here, we briefly review KPCA and fix the notation to be used. We are given a data set $(x_i)_{i=1}^n$ in $\mathcal{X}$. Also given is a Hilbert space $\mathcal{H}$, associated with a feature map $\phi : \mathcal{X} \to \mathcal{H}$ and positive definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Assume that the data has been centered, so that $\sum \phi(x_i) = 0$.

For any $w \in \mathcal{H}$, denote by $w^\sharp$ its adjoint, that is, the linear functional $v \mapsto \langle w, v \rangle$. This allows us to accommodate the case when $\mathcal{H}$ is infinite-dimensional: in general, the covariance is an operator

$$\Sigma = \frac{1}{n} \sum \phi(x_i)\phi(x_i)^\sharp.$$

KPCA is the problem of finding s directions in the Hilbert space such that the projected data has maximal variance. We can write this as the problem

$$\sup_{W \in \mathcal{S}_{\mathcal{H}}^s} \frac{1}{2} ||\Gamma W||_F^2. \qquad \textbf{(KPCA Problem)}$$

Here, $\mathcal{S}_{\mathcal{H}}^s$ is the set of vectors $W \in \mathcal{H}^s$ with orthonormal components. $\Gamma : \mathcal{H}^s \to \mathbb{R}^{n \times s}$ is an operator that maps such a W to an $n \times s$ matrix with components $\langle \phi(x_i), w_j \rangle$; so, it contains the projections of the features on the orthonormal frame W. The norm above is the usual Frobenius norm $||A||_F = \sqrt{\sum_{i,j} a_{ij}}$.

The standard method of carrying out KPCA relies on the Singular Value Decomposition. This algorithm takes $O(n^3)$ time, which is prohibitive for large data sets. In the next section, we present a refomulation that speeds this up.

3. Reformulation as the dual of a DC problem

Let $f = \frac{1}{2}||\cdot||_F$ and $g = \mathbb{1}_{\mathcal{S}^s_{\mathcal{H}}}$. (Throughout, we use the convention of convex optimization that indicator functions take on the values 0 and ∞.) **KPCA Problem** can then be written in the equivalent form

$$\inf_{W \in \mathcal{H}^s} g(W) - f(\Gamma W). \qquad \textbf{(DC Problem)}$$

This expression is a difference of convex functions, since norms are convex and indicator functions of convex sets are convex. ($\mathcal{S}^s_{\mathcal{H}}$ isn't convex, but as explained in [Ton+23], this can still be justified.) One way of solving such an optimization problem is by using the DC (Difference of Convex functions) algorithm. A simple example is shown in Figure 1 (code provided in the notebook).

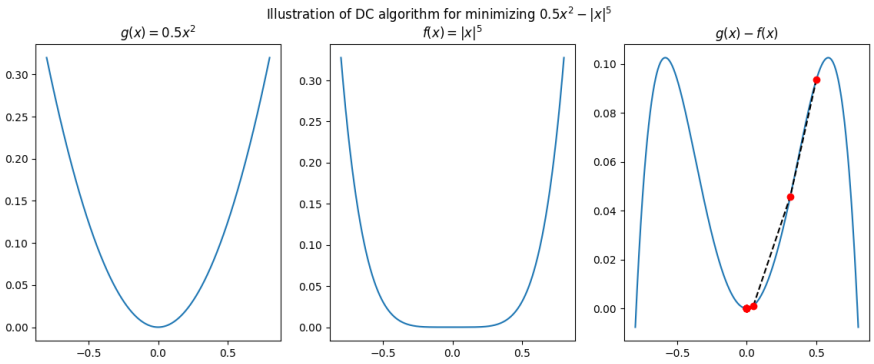

Figure 1. Illustration of the DC algorithm

By a theorem stated and proved in [Ton+23], we can instead solve the dual problem

$$\inf_{H \in \mathbb{R}^{n \times s}} f^*(H) - g^*(\Gamma^\sharp H), \qquad \textbf{(Dual Problem)}$$

which has the advantage that now the optimization problem is over the finite-dimensional space $\mathbb{R}^{n \times s}$. Since norms are self-dual, we have $f^* = f$. Some algebraic manipulations show that the second term can be simplified to $\pi(H) := \operatorname{Tr} \sqrt{H^T G H}$. Thus, we want to solve

$$\inf_{H \in \mathbb{R}^{n \times s}} \frac{1}{2}||H||_F^2 - \pi(H). \qquad (1)$$

This is the basic optimization problem considered in [Ton+23], together with extensions where the first term is modified. An algorithm that solves all of these will be stated later, but for now, we note that the second term is differentiable (as long as $H^T G H \succ 0$) with gradient

$$\nabla \pi(H) = G H U^T \operatorname{diag}\left(\frac{1}{\sqrt{\lambda(H^T G H)}}\right) U. \qquad (2)$$

Here, U is an orthogonal matrix obtained from the SVD of $H^T G H$, which is $s \times s$. As long as $s < n$, obtaining U will be much less expensive than performing the full SVD.

Having found a solution $\hat{H}$ to the dual problem (1), we can reconstruct the projections onto the principal components $\hat{W}$ using the following formula proved in [Ton+23]:

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

4. Extension to sparse or robust solutions

So far, $f = \frac{1}{2}||\cdot||_F$, so that solving **DC Problem** amounts to maximizing the variance. We will now modify the definition of f in order to enforce desirable properties on the solution. This will be done via an operation known as *infimal convolution*, which is a notion from convex analysis. The infimal convolution $f \,\square\, g$ of two functions is defined by

$$(f \,\square\, g)(x) = \inf\{f(x-y) + g(y) | y \in \mathbb{R}^n\}.$$

If we modify f to $f = \frac{1}{2}||\cdot||_F \,\square\, (\kappa||\cdot||)$, we get the *Huber objective*. The motivation for this definition is that $f^* = \frac{1}{2}||\cdot||_F + \mathbb{1}_{B_\kappa^*}$, where $\mathbb{1}_{B_\kappa^*}$ is the ball of radius κ with respect to the dual norm $||\cdot||_*$. So, if we solve **Dual Problem** with this f^*, the solution is constrained to a ball.

Alternatively, we can take $f = \frac{1}{2}||\cdot||_F \,\square\, \mathbb{1}_{B_\epsilon}$. This is known as the ϵ-*insensitive objective*. Then $f^* = \frac{1}{2}||\cdot||_F + \epsilon||\cdot||_*$. Solving **Dual Problem** with this choice of f, we get an additional regularization term $\epsilon||\cdot||_*$. The choice of this norm can lead to solutions with different properties. For example, by taking the ∞-norm and 1-norm (which are duals of each other), we get sparse solutions, as with the standard Lasso method.

5. Solving the dual problem

We briefly describe how **Dual Problem** is solved in the paper. This covers the KPCA case as well as the extensions covered in the previous section. As mentioned previously, we are trying to solve

$$\inf_{H \in \mathbb{R}^{n \times s}} f^*(H) - \pi(H),$$

and can proceed via the DC algorithm. This algorithm starts from some H_0 and then iterates the following for $k \geq 0$:

$$\begin{aligned} Y_k &\in \partial \pi(H_k) \\ H_{k+1} &\in \partial f(Y_k) \end{aligned} \tag{3}$$

The algorithm has been stated in terms of subgradients. Fortunately, this is not needed in this case; we have already noted that π is differentiable, with gradient (2). It terms out that $f = \frac{1}{2}||\cdot||_F \,\square\, \Psi$ is also differentiable, with gradient

$$\nabla\left(\frac{1}{2}||\cdot||_F \,\square\, \Psi\right)(Y) = \operatorname{prox}_{\Psi^*}(Y) \tag{4}$$

The right side of this equation is the *proximal operator* (another notion from convex analysis), defined by

$$\text{prox}_f(v) = \arg\min_x f(x) + \frac{1}{2}||x - v||^2.$$

In summary, **Dual Problem** is solved by iterating (3) based on the equations (2) and (4).

6. Numerical experiments

6.1. Comparison analysis

Nyström approximation

The paper focuses on evaluating the performance of its novel approach by solving the dual objective function using the Limited-memory BFGS (L-BFGS) quasi-Newton method, known for its efficiency in converging to critical points.

The study considers several solvers for comparison: the Lanczos method, Randomized Singular Value Decomposition (RSVD), and Full SVD, the latter serving as a baseline metric.

Additionally, we compare the proposed approach with a relevant solver, the Nyström approximation, commonly used to avoid the computation of the full Gram matrix, which has a size of $n \times n$.

Recall on best rank-p approximation : Considering the eigendecomposition of the Gram matrix K:

$$K = U\Lambda U^T$$

with $U = [\mathbf{u}_1, \ldots, \mathbf{u}_n]^T$ the matrix of the eigenvectors, and $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$, the diagonal matrix of the eigenvalues ($\lambda_1 \geq \lambda_2 \cdots \geq \lambda_n$).

Then, an approximation of K of rank p is given by considering the first eigenvalues and eigenvectors: it is the best rank-p approximation of the initial matrix.

$$K_p = U_p \Lambda_p U_p^T$$

Solving Kernel-PCA consists in finding the best rank-p approximation of the gram matrix which is known to be computationally expensive of order $O(n^3)$.

Therefore, Nyström approximation proposes an algorithm to reduce the cost by approximating a kernel matrix by selecting a random subset of its columns and using those to reconstruct the full matrix :

Data : $n \times n$ Gram matrix G, $\{p_i\}_{i=1}^n$ such that $\sum_{i=1}^n p_i = 1$, $c \leq n$, and $k \leq c$.
Result : $n \times n$ matrix $\tilde{G}$.
- Pick c columns of G in i.i.d. trials, with replacement and with respect to the probabilities $\{p_i\}_{i=1}^n$; let I be the set of indices of the sampled columns.
- Scale each sampled column (whose index is $i \in I$) by dividing its elements by $\sqrt{cp_i}$; let C be the $n \times c$ matrix containing the sampled columns rescaled in this manner.
- Let W be the $c \times c$ submatrix of G whose entries are $G_{ij}/(c\sqrt{p_i p_j}), i \in I, j \in I$.
- Compute W_k, the best rank-k approximation to W.
- Return $\tilde{G}_k = CW_k^+ C^T$.

Figure 2. Nyström approximation algorithm [DM05]

The computational cost is then of order $O(c^3 + cnk)$.

Figure 3 provides an example of the resulting projection on the approximated principal components by nyström approximation for random dataset $d = 2, n = 500$ by choosing an discrete uniform distribution over $\{1, .., n\}$, with probability $p_i = 1/n$.

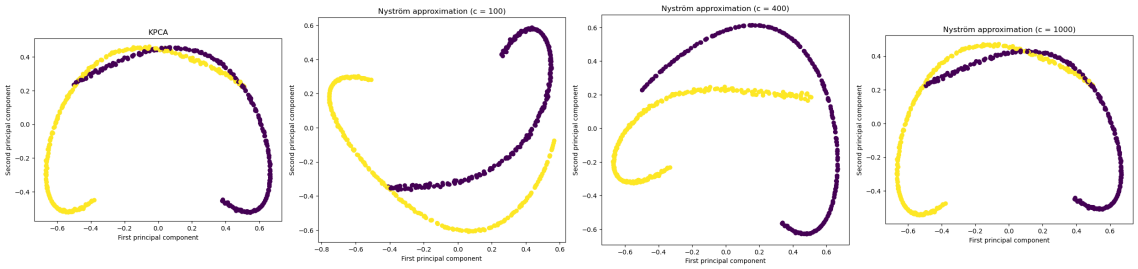

Figure 3. Nyström approximation for different values of c

Results of comparison analysis

In our analysis we compared the performance of the proposed L-BFGS-based algorithm for solving the dual objective function with three other solvers : Full Singular-Value-Decomposition (Full SVD), Randomized SVD, and Lanczos as proposed in the paper. The nystrôm approximation algorithm with $c = 100$ was added as new solver to compare the performance.

Full SVD serves as the reference metric to measure the perfomance of the three other solvers with the optimal solution $d_{opt} = -\frac{1}{2}\sum_{i=1}^{n} \lambda_i$ where λ_i represents the i-th largest eigenvalue of the symmetric gram matrix of the choosen kernel.

Therefore error of each solver is the relative distance to the optimal solution defined by $\eta = \frac{|d(\hat{H}) - d_{opt}|}{d_{opt}}$ where $d(\hat{H})$ is the dual cost equation (1), of the approximated principal components matrix $\hat{H}$ computed by the solvers.

Figure (3) shows the resulting computation time in seconds and the relative error for four different size $N^2 = \{128^2, 512^2, 1024^2, 2048^2\}$ of the kernel gram matrix and for a fixed number of feature $d = 2$, the kernel chosen here is the laplacien kernel.

It is notable that Nyström approximation has the lower computational time among four other algorithms which is traded by less precision as n becomes larger. The proposed LBFGS method for the solving the **Dual Problem** has the highest computational time when the number of observations n is low but becomes faster as n increases. As n becomes larger the proposed LBFGS approach becomes the more and more reliable in terms of computational time and precision.

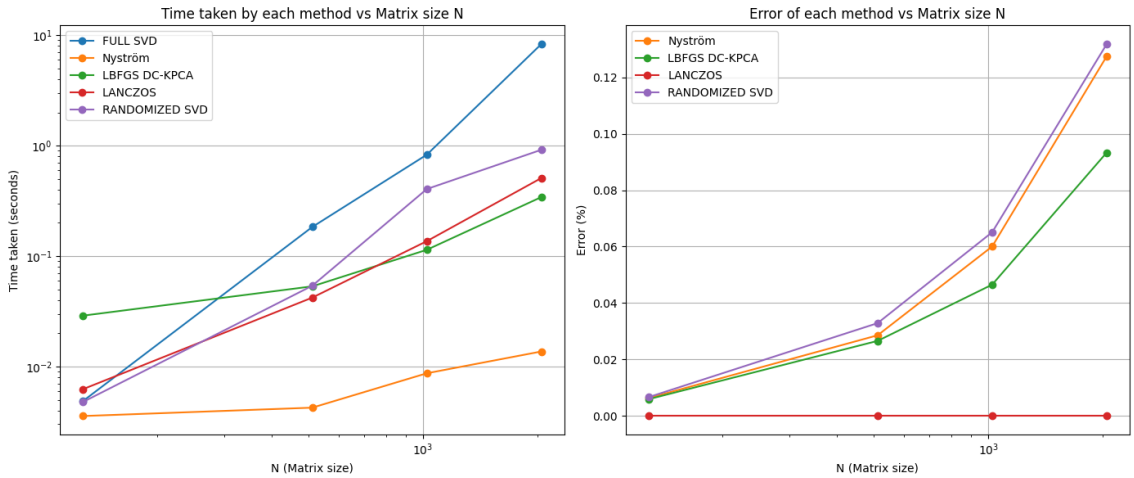

Figure 4. Running time and error for different solution methods to KPCA for four different value of n.

6.2. Huber and ϵ-insensitive objective

A numerical experiment was carried out to show the effect of the Huber objective on solutions to the **Dual Problem**. We generated $n = 1000$ samples from a standard normal distribution in $\mathbb{R}^{10}$, and take $s = 2$ so that the solutions can be plotted.

The dual solution $\hat{H}$ was computed according to the method in this paper, taking $\kappa = 0.1$ in the Huber objective. The results appear in Figure 5, and show the solution constrained to a ball of radius κ, as expected from Section 4.

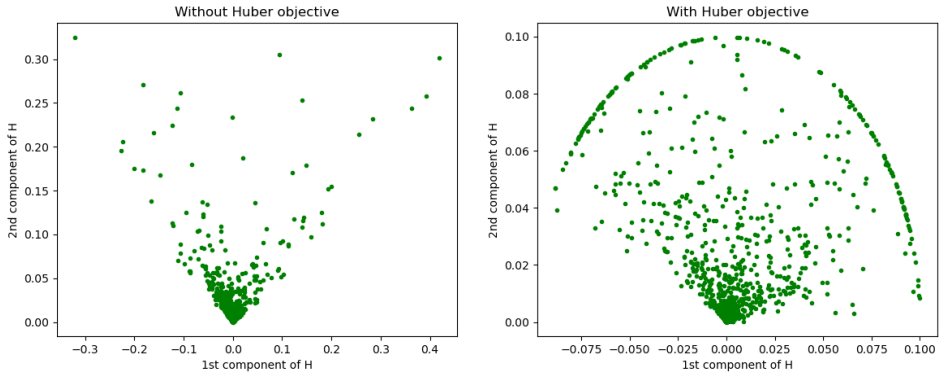

Figure 5. Effect of Huber objective on solution to the dual problem

The effect of the ϵ-insensitive objective producing sparsity penalty is illustrated in Figure 6.

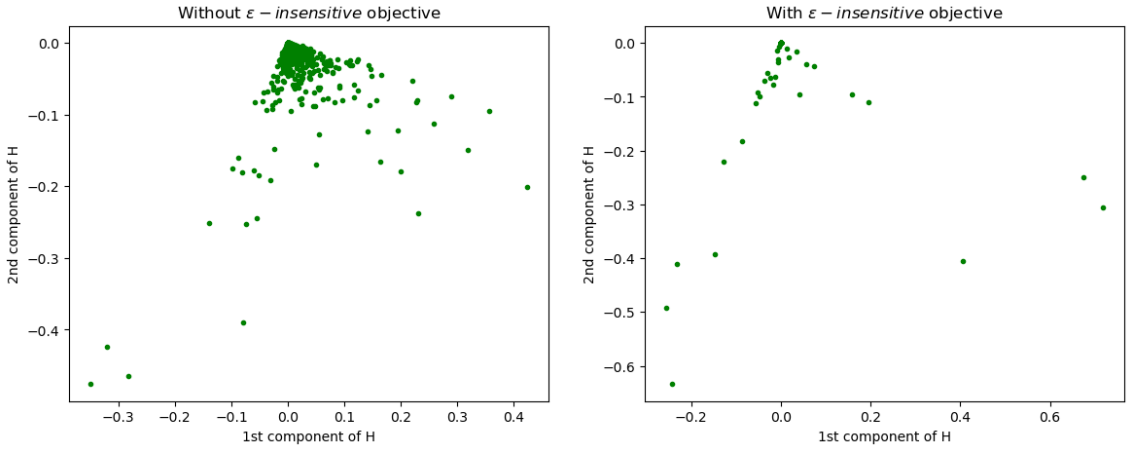

Figure 6. Effect of ϵ-insensitive objective on solution to the dual problem

7. Critical evaluation of the methods presented

7.1. Pros

- The first main benefit of the method presented in this paper is its computational advantage over the full SVD (and several other methods for performing KPCA).
- The second main benefit is the ability to incorporate desirable properties such as sparsity and robustness into the solution via straightforward modifications of the objective function.
- KPCA is studied in the context of convex analysis. This brings many interesting theoretical tools to bear on the problem - in the paper, ideas such as infimal convolutions and proximal operators play an important role. The infimal convolution allows us to extend the objective function, and its gradient can be expressed as a proximal operator.
- The idea of extending the objective function via the infimal convolution is very flexible - in the paper, only two instances of this are considered, yielding robustness and sparsity. In fact, "the DCA algorithm can be applied as long as the computation of prox_{Ψ^*} is possible" ([Ton+23]).

7.2. Cons

- Suppose that the number of principal components sought s is of the same order as n, the size of the data set. Then the time complexity of the methods in this paper are $O(n^3)$, which is the same as for the SVD. Thus, in this case, we expect there to be no real advantage.
- In theory, the optimization problems considered in the paper can be slightly modified in order to be convex, as the constraint "...can be relaxed to the convex hull of the Stiefel manifold as the solutions necessarily lie on the boundary." ([Ton+23]). However, in practice **Dual Problem** is nonconvex and local minima or saddle points can occur.

8. Conclusion

In summary, the paper under consideration takes a standard machine learning method (KPCA) and studies it in the framework of DC algorithms. This had been done before

in [BT21] for PCA, but not for KPCA. The paper presents numerical evidence that the resulting algorithm is faster than other standard methods of performing KPCA. In addition, infimal convolutions can be used to modify the objective function in order to obtain sparsity or robustness.

One further line of research might be to search for other desirable properties that can be obtained via infimal convolution of the objective function. Another possibility would be to study the convergence properties of the DC algorithm, since, as noted before, it converges to a critical point, but not necessarily a minimum, of **Dual Problem**.

References

[BT21] Amir Beck and Marc Teboulle. "Dual Randomized Coordinate Descent Method for Solving a Class of Nonconvex Problems". In: *SIAM Journal on Optimization* 31.3 (2021), pp. 1877–1896. DOI: 10.1137/20M133926X.

[DM05] Petros Drineas and Michael W. Mahoney. "On the Nystrom Method for Approximating a Gram Matrix for Improved Kernel-Based Learning". In: *Journal of Machine Learning Research* 6.72 (2005), pp. 2153–2175. URL: http://jmlr.org/papers/v6/drineas05a.html.

[Ton+23] Francesco Tonin et al. *Extending Kernel PCA through Dualization: Sparsity, Robustness and Fast Algorithms*. 2023. arXiv: 2306.05815 [cs.LG]. URL: https://arxiv.org/abs/2306.05815.