Journal of Machine Learning Research 18 (2018) 1-43 Submitted 3/16; Revised 1/18; Published 4/18

Compact Convex Projections

Steffen Grünewälder S.GRUNEWALDER@LANCASTER.AC.UK
Department of Mathematics and Statistics
Lancaster University
Lancaster, England

Editor: Mark Schmidt

Abstract

We study the usefulness of conditional gradient like methods for determining projections onto convex sets, in particular, projections onto naturally arising convex sets in reproducing kernel Hilbert spaces. Our work is motivated by the recently introduced kernel herding algorithm which is closely related to the Conditional Gradient Method (CGM). It is known that the herding algorithm converges with a rate of $1/t$, where t counts the number of iterations, when a point in the interior of a convex set is approximated. We generalize this result and we provide a necessary and sufficient condition for the algorithm to approximate projections with a rate of $1/t$. The CGM, which is in general vastly superior to the herding algorithm, achieves only an inferior rate of $1/\sqrt{t}$ in this setting. We study the usefulness of such projection algorithms further by exploring ways to use these for solving concrete machine learning problems. In particular, we derive non-parametric regression algorithms which use at their core a slightly modified kernel herding algorithm to determine projections. We derive bounds to control approximation errors of these methods and we demonstrate via experiments that the developed regressors are en-par with state-of-the-art regression algorithms for large scale problems.

1. Introduction

Convex sets and projections onto convex sets are omnipresent in Machine Learning and Statistics. Projections appear already in the most basic approaches like in ordinary least squares regression where the the estimate can be interpreted as the projection of some target vector onto a linear subspace. By adding constraints one arrives naturally at a convex projection problem. Similarly, the ridge regressor and Gaussian process regressor are closely related to projections onto balls and the lasso estimator (Tibshirani, 1996) corresponds to a projection onto a simplex (again a convex set). Regularization approaches with convex constraints are, in general, closely related to the problem of determining a projection. For example, sparsity constraints are often enforced by optimizing over the standard simplex and algorithms that determine projections onto the simplex have been studied extensively (Duchi, Shalev-Shwartz, Singer, and Chandra, 2008).

This paper is about exploring the usefulness of some closely related algorithms for determining projections onto convex sets in the large data context. Our inspiration for this line of research dates back to the paper by Welling (2009) in which an algorithm, called the herding algorithm, has been introduced which computes compact representations of probability measures. By a compact representation we mean here a representation that is supported on few points of the sample space and that can be used to calculate efficiently expectations of functions. The algorithm has garnered attention in recent years and various generalizations of the method have been explored. In one of

these works it has been studied how the algorithm behaves in a non-parametric setting where a kernel function is defined on the sample space (Chen, Welling, and Smola, 2010). The authors analyzed the interplay between the algorithm and the reproducing kernel Hilbert space (RKHS, Aronszajn (1950)) associated with the kernel function. Their main finding has been that the algorithm has a guaranteed rate of convergence towards the given probability measure (or, more accurately, towards its representer in the RKHS) of $1/t$, where t is the number of iterations for which the algorithm is run. The rate of $1/t$ is an improvement over randomly selected support points, which would guarantee a rate of $1/\sqrt{t}$, but the rate is slow compared to what many other numerical algorithms yield. It is worth pointing out that we consider here convergence in $\|\cdot\|$, while in the literature results are often reported for quadratic objectives like $\|\cdot\|^2$. The virtue of this algorithm is that the computational costs per iteration are linear in the sample size. These low costs in dependence of the sample size make this algorithm, and related algorithms, attractive in the large data context where algorithms with faster rates of convergence are often prohibitively expensive to compute.

The herding algorithm is iterative and uses at its core an inner product between the approximation error $y - p_t$ at iteration t and a set of candidates S that can be chosen to reduce the error. Here, y lies inside a convex set C, p_t is the approximation of y at iteration t and S is a subset of C. The algorithm selects elements which maximize this inner product $\langle y - p_t, x\rangle$ over $x \in S$. We asked ourselves somewhat naïvely what would happen if y lies outside of C. One would hope that the algorithm converges in this case to the point in C that is closest to y, or in other words, that it would converge to the projection Py of y onto C. The first observation we had was that this will, in fact, be true if $y - Py$ stands orthogonal on the (affine) subspace spanned by C and if Py lies in the (affine) interior of C, because in this case the algorithm is completely unaffected by $y - Py$ and it behaves equivalently to the case where we would apply it directly to $Py \in C$. In this setting, the standard guarantees tell us that the algorithm converges with a rate of $1/t$ to the projection. Our initial observation was in a way naïve since the literature on herding provides a way to show that the algorithm with an added line search converges to the projection. The argument to verify the convergence is based on the observation from Bach, Lacoste-Julien, and Obozinski (2012) that the herding algorithm with an added line search is equivalent to a well known method called the conditional gradient method (CGM, Frank and Wolfe (1956)). Basic guarantees on the convergence of the CGM imply directly that the method converges with a rate of $1/\sqrt{t}$ to the projection (see our Literature Section below for more details). This convergence result is reaffirming, however, the rate of convergence the result guarantees is a slow rate of $1/\sqrt{t}$ and not $1/t$.

The rate of $1/\sqrt{t}$ is, in fact, the actual rate with which CGM converges in non-trivial projection problems. The reason for this is that, in interesting cases, projections lie in the boundary of the convex set and CGM itself achieves only in exceptional cases a rate that is better than $1/\sqrt{t}$ if the solution of an optimization problem lies in the boundary (Cannon and Cullum, 1968). In recent years extensions of the basic CGM algorithm have become popular and it was shown that a linear rate of convergence (a rate significantly better than $1/t$) is achieved by a particular extension, the CGM with away steps algorithm, even if the solution lies in the boundary (Lacoste-Julien and Jaggi, 2015). This is a significant result, but it comes with a caveat: we gain strong guarantees only if the convex set has a large pyramidal width. For instance, the d dimensional cube has a large pyramidal width and the authors provide an upper bound on the reduction of the approximation error in the order of $1/d^2$ per step. The convex sets we are mainly interested in are of the form $C = \{k(x,\cdot) : x \in X\} \subset \mathcal{H}$, where k is the reproducing kernel of the RKHS $\mathcal{H}$. Here, the dimension of the spanned subspace is often equal to the sample size (when using a Gaussian kernel,

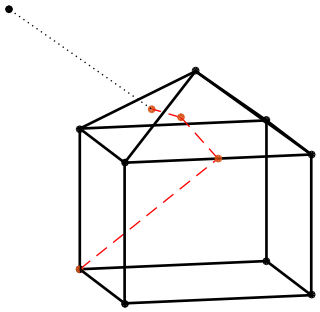

Figure 1: An illustration of the application of the herding algorithm to a projection problem. The aim is to find the projection of a ray starting at the black dot onto the house. The house is compact and convex. The herding algorithm finds for such sets approximations of the projection with an accuracy in the order of $1/t$, where t is the number of iterations. The red dots show the approximation after $0, 2, 5$ and 100 steps with an initial estimate that equals the North-West corner on the bottom of the house.

for example). Furthermore, C is significantly more complicated than a cube and we expect that the upper bound gives us guarantees on the reduction that are significantly worse than $1/n^2$ per iteration, where n is the size of the sample. In the large data context, where $n \geq 10^5$, these reductions per step are tiny. Another recent approach to improve the convergence behavior of the CGM has been developed in Garber and Hazan (2013). Standard CGM is in this approach combined with what the authors call a Local Linear Optimization Oracle (LLOO) and it is shown that a linear rate of convergence is achieved by their method. Like in the away step approach the geometry of the convex set factors into the performance. For the method from Garber and Hazan (2013) the geometry affects the run-time of the LLOO and, hence, the run-time of the CGM with the added LLOO. The approach works well for simple shaped convex sets like the simplex but becomes difficult when C has no particular structure as in our RKHS setting. Understanding better the relevant geometric properties of C with respect to the reproducing kernel is an intriguing problem, but it is beyond the aims of this paper. We aim here for a deeper understanding of the core algorithms and their interplay with the projection problem in Hilbert space. The following figure visualizes the relation between the different CGM like methods. We focus in this paper on the methods in the shaded area.

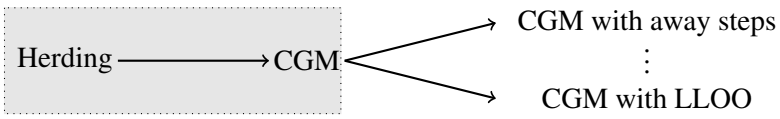

Figure 2: The relation between various CGM like methods.

CGM is known to converge with a linear rate if there exists a ball around the solution inside the convex set (this is also known as Slater's condition, Beck and Teboulle (2004)). Similarly, for the herding algorithm it has been shown that this very same condition guarantees a rate of $1/t$ (Chen et al., 2010). To the best of our knowledge, both convergence results have been derived

independently of each other and it is telling that in both approaches the same assumption plays a fundamental role (the existence of a ball around the optimum). The assumption appears naturally in the analyses since it allows a strong reduction of estimation errors per iteration independently of the direction in which the errors point. If this assumption does not hold then there are directions which pose problems, i.e. if an error builds up in such a direction then these errors will only decay slowly over time. This is why solutions that lie in the boundary pose problems: errors that point away from the convex set can not be easily reduced. We refine the convergence argument to be able to deal with such errors. On a high level our approach can be described in the following way: consider a convex set with finitely many extremes then each point in the boundary is in the relative interior of a face of the original convex set. For example, in Figure 1 our convex set is the house and we want to compute the projection of the black dot onto the house. The dotted line visualizes the process of projecting onto the house and the red dot at the end of the line is the projection that we want to determine algorithmically. The projection does not lie in the interior of the house but it lies inside a face of the convex set (the triangle that contains the projection). A face of a convex set is in turn a convex set and if we would apply the algorithm directly to this face then we could show fast convergence. In general, we can not identify the face that contains the projection without using significant computational resources. However, we can treat elements outside of this face, which are chosen by the algorithm, as perturbations that disturb the behavior of the algorithm. For instance, we applied the herding algorithm to the problem in Figure 1 and we used an initial approximation of the projection which corresponds to the North-West corner on the bottom of the house. The dashed line in red shows how the approximation evolves over multiple steps and how the error that is introduced by the initialization shrinks over time. Our line of attack in the theoretical part of this paper is to control such perturbations and to show that the rate of convergence of the herding algorithm is not negatively affected by these.

We were also interested in understanding how CGM like algorithms to determine projections in Hilbert spaces can be exploited in statistical problems. A blueprint is given in the paper by Chen et al. (2010) where the herding algorithm is applied in an RKHS $\mathcal{H}$ to approximate elements inside a compact and convex set $C \subset \mathcal{H}$. It is natural to consider extensions in which C is modified in a suitable way to represent a space of solutions of certain statistical problems. We showcase this approach in the regression problem. In this case, C is the space of regression functions. The projection approach itself is generic and one can gain with relative ease algorithms that infer kernel regressors from data. We call these CCP-regressors where the acronym CCP stands for compact convex projection. The advantage of this approach is that the corresponding algorithms are cheap to compute and they are applicable in the large data context. We find in experiments that the algorithms we derive are en-par with established kernel regression algorithms like the Gaussian process or the fast kernel ridge regressor (FastKRR, Zhang, Duchi, and Wainwright (2013)).

1.1 Literature

We aim in this section for a short overview summarizing key results that put our work in perspective. We start by stating the herding algorithm for approximating a point x^* inside a convex set C that is induced by the finite set of points S, i.e. $C = \operatorname{cch} S$, where cch denotes the closed convex hull operator. C is typically a subset of $\mathbb{R}^d$ equipped with the Euclidean inner product or some other Hilbert space. Given the starting value $w_0 = x^*$ and the initialization $t = 1$ the herding algorithm iterates

1. choose $x_t \in \arg\max_{x \in S} \langle w_{t-1}, x \rangle$,
2. set $w_t = w_{t-1} - (x_t - x^*)$,
3. set $t \leftarrow t + 1$.

This basic routine is combined with some termination criterion like running the algorithm for T iterations. The approximation of x^* is then $(x_1 + \ldots + x_T)/T$ where T denotes here the number of iterations the algorithm was run for.

The herding algorithm is a special case of what is called the *conditional gradient method (CGM)* or the *Frank-Wolfe algorithm* (Frank and Wolfe, 1956; Bach et al., 2012). The conditional gradient method is known since the fifties and various results about its convergence behavior have been derived throughout the years. The CGM method tries to approximate the minimal value of a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ on a set C by iterating two steps after an initialization with some value $x_0 \in C$:

1. Choose an $x^\star \in \arg\min_{x \in C} \langle x - x_{t-1}, \nabla f(x_{t-1}) \rangle$,
2. perform a line search over $\lambda \in [0, 1]$, i.e. choose a

$$\lambda^\star \in \operatorname*{arg\,min}_{\lambda \in [0,1]} f(x_{t-1} + \lambda(x^\star - x_{t-1}))$$

   and update $x_t = x_{t-1} + \lambda^\star(x^\star - x_{t-1})$.

Standard results for the CGM method hold for continuously differentiable functions on convex compact subsets C of $\mathbb{R}^d$ (Beck and Teboulle, 2004). Let $f^* = \min_{x \in C} f(x)$ then it is known that a sequence $\{x_t\}_{t \geq 0}$ produced by the CGM attains the minimum eventually, i.e. $\lim_{t \to \infty} f(x_t) = f^*$ and there exists a constant b such that $f(x_t) - f^* \leq b/t$. A differentiable function on a compact set is Lipschitz continuous and the constant b depends on the Lipschitz constant and on the diameter of the set C, $\operatorname{diam}(C) = \sup_{x,y \in C} \|x - y\|$. We can use the CGM to find the projection of z onto C. Letting $f(x) = \|x - z\|^2$, with $\|\cdot\|$ the Euclidean norm, we can observe that f is continuously differentiable with derivative $2\langle x - z, \cdot \rangle$ and we are assured that $\|x_t - z\|^2 - \min_{x \in C} \|x - z\|^2 \leq b/t$ for some constant b and all $t \geq 0$. Observe that this rate is slower then what we aim for since we gain here a rate of about $t^{-1/2}$ for the convergence of $\|x_t - z\|$. The norm itself is not differentiable and we can not derive the faster rate of $1/t$ through these results.

It is also known that the rate of convergence can not be improved in general (Cannon and Cullum, 1968)[Thm. 2]. Hence, stronger assumptions are needed to gain faster rates of convergence. One typical assumption is that *Slater's condition* holds. Slater's condition is in our context the assumption that $z \in \operatorname{int} C$, where $\operatorname{int} C$ denotes the interior of C. Proposition 3.2 in Beck and Teboulle (2004) demonstrates that if Slater's condition is fulfilled then the CGM produces a sequence $\{x_t\}_{t \geq 0}$ which satisfies

$$\|x_t - z\| \leq be^{-tq/2}$$

for some positive constants b and q. The rate is significantly faster than what is known for the herding algorithm but the result is not applicable to the problem of computing projections since (non-trivial) projections lie in the boundary of C and not in the interior. These convergence results are nevertheless fundamental and they are a corner stone for various studies. For instance, in Jaggi (2013) these convergence results for the CGM are combined with a duality argument to give bounds

on the duality gap between the primal and dual solution of the optimization problem. Another line of research considers extensions of the CGM which are build to circumvent the sub-linear rate of convergence. We discussed already the most prominent approaches: the classical approach from Wolfe (1970) in which away steps are added to the CGM and the approach from Garber and Hazan (2013) which both guarantee a linear rate of convergence.

Much of the research on the CGM has been done in the general context where an arbitrary continuously differentiable function on a convex set is optimized. However, the more specialized problem of determining projections onto convex sets has also garnered attention. Especially, in the context of sparsity. The convex set C that is typically considered in this context is the standard simplex in $\mathbb{R}^d$ which is $\Delta^{d-1} = \operatorname{cch}\{e_n : 1 \leq n \leq d\}$, where $e_1, \ldots, e_d$ is the standard basis in $\mathbb{R}^d$. The simplex Δ^{d-1} is of particular importance because ℓ_1-constraints can be naturally expressed through it: $w \in \mathbb{R}^d_+$ with $\|w\|_{\ell_1} = 1$ implies that $w = \sum_{i=1}^d \langle w, e_i \rangle e_i \in \operatorname{cch}\{e_n : 1 \leq n \leq d\} = \Delta^{d-1}$ (and vice versa). The CGM acting on the simplex can be used to find sparse solutions for a variety of problems. Most notably the *lasso estimator* (Tibshirani, 1996) can be found through an application of the CGM to a simplex (Clarkson, 2010). In Duchi et al. (2008) a similar problem is studied. The motivation are there algorithms which require projections onto a simplex. The authors develop a projection algorithm that can be computed in the order of d operations on average (the algorithm is stochastic) where d is the dimension of the simplex. This resembles the order of run time of the CGM.

Another set of important results for the CGM rely on strong convexity assumptions about the function f and the set C, see for exampleGarber and Hazan (2015) and references therein. Under such assumptions one can obtain a rate of $1/t$ for finding the projection on C, i.e. the rate of convergence applies independently of where in C the solution lies. The boundary of a strongly convex set C bends significantly. In our paper we assume very different properties of C. We are interested in the convex hull C of a finite set S and faces of such sets C are flat, i.e. they do not bend.

Closely related to the closed convex hull of a set S is the minimum enclosing ball of S (MEB, Clarkson (2010)). The difference between the two is that an Euclidean ball bends (the Euclidean norm is uniformly convex) while the closed convex hull interpolates the boundaries through hyperplanes. For example, the MEB of $S = \{\pm e_1, \ldots, \pm e_d\}$ is the Euclidean unit ball while $\operatorname{cch} S$ is an ℓ_1-ball (a diamond). There exist efficient algorithms for determining the MEB under different conditions. Furthermore, a variety of machine learning problems can be rephrased as the problem of finding an MEB (support vector algorithms are approached in this way in Tsang et al. (2005a,b)).

1.2 Contributions

In this paper we study the usefulness of conditional gradient methods to determine projections onto compact convex sets in Hilbert spaces. The contributions split naturally into theoretical and applied contributions through which we explore this theme. The theoretical contributions are *two theorems*. Our *first theorem* shows that the herding algorithm retains its rate of convergence of $1/t$ if, and only if, the perturbations introduced through points outside the minimal face are coupled in a certain way. The approach we use (considering the minimal face to which well-known results can be easily adapted and controlling perturbations which are introduced by elements outside of the minimal face) is to the best of our knowledge new and we expect that this line of reasoning can be extended in the future to study more advanced methods like the CGM with away steps.

Our *second theorem* analyses why the standard CGM fares worse in this setting than the herding algorithm. On the applied side we show how projections onto compact convex sets in kernel space can be exploited to develop novel machine learning algorithms. We demonstrate this by developing a *new and fast kernel regressor* that is en-par with state-of-the-art non-parametric regressors. We provide *approximation error bounds* for the method by means of a dual approach and we study its performance in experiments. The approximation error bound is a novel modification of an approach from Wolfe (1976) which sacrifices tightness in favor of a significantly reduced computation time.

2. Compact Convex Projections

We use the following algorithm to find the projection of $y \in \mathcal{H}$ onto the compact convex set $C \subset \mathcal{H}$ in terms of a set of support points from the compact set $S \subseteq C$ which contains the extremes of C. In the following we denote the extremes of C with $\operatorname{ex} C$, i.e. $\operatorname{ex} C = \{x : x \in C, \text{ there exists no } y, z \in C, \alpha \in (0,1) \text{ such that } x = \alpha y + (1-\alpha)z\}$.

Algorithm 1. Input: $y \in \mathcal{H}$, $T \in \mathbb{N}$, and $S \subset \mathcal{H}$.

1. [Initialize] Set $w_0 = y, t = 1$.
2. [Optimization oracle] Choose $x_t \in \arg\max_{x \in S} \langle w_{t-1}, x \rangle$.
3. [Update weight] Set $w_t = w_{t-1} - (x_t - y)$.
4. [Iterate] If $t < T$ then increment t by 1 and go back to **2**.
5. [Terminate] Return the approximation $(x_1 + \ldots + x_T)/T$.

A maximizer exists in Step 2 because we search for a maximum of a continuous function on a compact set. If there are multiple maximizer then we can choose an arbitrary one, for instance, we can enumerate S and choose the maximizer with the lowest index. This is the herding algorithm as stated in Section 1.1 with the only modification being that it is applied to y which can lie outside of C. The weight at step t is essentially the approximation error scaled up by t because $w_t = (t+1)y - (x_1 + \ldots + x_t) \approx t(y - p_t)$, where $p_t = (x_1 + \ldots + x_t)/t$ is the approximation. We can also add a line search to the optimization:

Algorithm 1 (ls). Input: $y \in \mathcal{H}$, $T \in \mathbb{N}$, and $S \subset \mathcal{H}$.

1. [Initialize] Set $w_0 = y, t = 1$.
2. [Optimization oracle] Choose $x_t \in \arg\max_{x \in S} \langle w_{t-1}, x \rangle$.
3. [Line search] Calculate $\tilde{\alpha}_t = \langle w_{t-1}, w_{t-1} + (x_t - y) \rangle \,/\, \|w_{t-1} + (x_t - y)\|^2$.
4. [Line search] If $t = 1$ set $\alpha_t = 1$, otherwise set $\alpha_t = 1 \wedge \tilde{\alpha}_t$.
5. [Update weight] Set $w_t = (1 - \alpha_t) w_{t-1} - \alpha_t (x_t - y)$.
6. [Iterate] If $t < T$ then increment t by 1 and go back to **2**.
7. [Terminate] Set $\beta_i = \alpha_i \prod_{j=i+1}^{T} (1 - \alpha_j)$ for all $i \leq T$ and return the approximation $\beta_1 x_1 + \ldots + \beta_T x_T$.

We use here the convention that $0/0 = 1$ which implies that $\alpha_t = 1$ if $x_t = p_t$. The weight in the

line search algorithm is similar to the weight of the herding algorithm scaled down by $1/t$, i.e. the weight in the line search algorithm is $w_t = y-(\beta_1^t x_1+\ldots+\beta_t^t x_t)$ where $\beta_i^t = \alpha_i \prod_{j=i+1}^t (1-\alpha_j)$. The choice $\tilde{\alpha}_t = \langle w_{t-1}, w_{t-1} + (x_t - y)\rangle \,/\, \|w_{t-1} + (x_t - y)\|^2$ minimizes $\|w_t\|$ over all choices of $\tilde{\alpha}_t \in \mathbb{R}$. We need to be assured that the scaling α_t lies in the interval $[0, 1]$ to guarantee that we have a convex combination of points of S as the approximation. $\tilde{\alpha}_t$ is always non-negative since for $t \geq 1$

$$\langle w_{t-1}, w_{t-1} + (x_t - y)\rangle = \langle w_{t-1}, x_t\rangle - \sum_{i=1}^{t-1} \beta_i^{t-1} \langle w_{t-1}, x_i\rangle \geq \langle w_{t-1}, x_t\rangle - \max_{x\in S} \langle w_{t-1}, x\rangle = 0,$$

where we define a sum from $i = 1$ to 0 to be 0. The inequality above holds because the β's are non-negative and they sum to 1. $\tilde{\alpha}_t$ can, however, be strictly larger than 1. Hence, to guarantee that our approximation is a convex combination of points from S we need to force the scaling factor back into the interval $[0, 1]$. We do this in the algorithm by assigning the value 1 to the scaling factor in this case. This choice minimizes $\|w_t\|$ because the derivative of $\|w_t\|^2$ with respect to α is non-positive for all $\alpha \leq \tilde{\alpha}$, that is

$$2(\alpha \|w_{t-1} + (x_t - y)\|^2 - \langle w_{t-1}, w_{t-1} + (x_t - y)\rangle) \leq 0, \quad \text{for all } \alpha \in (-\infty, \tilde{\alpha}_t].$$

Note that Algorithm 1 is the herding algorithm applied to a point outside the convex set. This corresponds to the CGM where the optimization over the step size is replaced by a step size of $1/t$. Algorithm 1 (ls) is the CGM algorithm applied to the problem of finding the projection of y on the convex set spanned by S.

2.1 Rate of Convergence

We take a closer look at the convergence behavior of the herding algorithm and the standard CGM when applied to a projection problem. It is known that the herding algorithm converges with a rate of $1/t$ to elements in the interior of compact convex sets in Hilbert space (this is shown in Chen et al. (2010) for what is called the marginal polytope. The same proof works for arbitrary compact convex sets in a Hilbert space). Hence, if we want to approximate the projection of an element y onto the convex set C and y is already in the interior of C (in essence a trivial projection problem) then standard proofs guarantee a rate of $1/t$. Obviously, the interesting case is one where the projection lies in the boundary. We provide a theorem below (with the proof being postponed to Section 6.1, p. 26) which extends current results to the case where y lies in the boundary or outside the convex set if an assumption on the perturbations is fulfilled. We also show that this assumption is both necessary and sufficient for the algorithm to converge with a rate of $1/t$. We conclude this section by presenting a number of settings in which this assumption is fulfilled.

It is instructive to go through the main ideas of our approach. Consider again Figure 1. We can observe that there is a minimal face of the convex set which contains the projection Py of y onto the convex set, i.e. there exists a face that contains Py and is a subset of any other face that contains Py. In the figure, the minimal face is the convex set which contains the projection, i.e. the red dot. The vector $y - Py$, with $Py \in C$ being the projection, stands orthogonal on this minimal face. Furthermore, this minimal face is either an extreme point or Py lies in the relative interior of it. In the latter case we have a ball around Py (relative to the affine subspace spanned by the minimal face) which is contained in the minimal face. This property of the existence of a ball

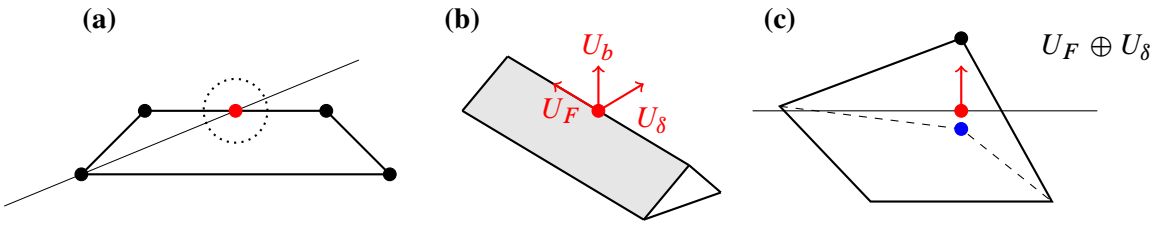

Figure 3: (a) A visualization of Lemma 1. (b) The figure depicts a triangular prism. The point of projection Py is marked by the red dot. The minimal face that contains Py is the top edge of the prism. The red arrows visualize the three subspaces U_F, U_b and U_δ. (c) The figure shows the projection of a set C_c onto the subspace $U_F \oplus U_\delta$ (the polytope enclosed by the thick line). The red dot marks the projection of Py onto the subspace and the red arrow visualizes the error at some stage t, i.e. w_t. The horizontal line represents $\{x : \langle w_t, x\rangle = 0\}$. The inner product between the black dot and w_t might be very different if evaluated over the whole space, i.e. if the component $P_b w_t$ is considered too. In particular, the polytope might appear to the algorithm like the modified polytope in which the black dot is replaced by the blue dot. In this modified polytope the best aligned element is the leftmost point.

around the element that we want to approximate is of crucial importance in all current approaches that demonstrate a fast reduction of error. We can therefore hope for a fast reduction of error in the affine subspace spanned by F. In fact, since $y - Py$ stands orthogonal on the minimal face, we can also observe that the algorithm applied to the minimal face behaves in exactly the same way as if we run it with Py instead of y and we can conclude that the algorithm converges with a rate of $1/t$ if it chooses only elements of F. The obvious problem is that we do not know the minimal face beforehand and we can not force the algorithm to choose only elements from it. At this point, a difficult task is to measure how much harm elements outside the minimal face can do when they are chosen by the algorithm. One important observation here is that for any element x outside the minimal face there exists no ball B around Py such that $\operatorname{aff}\{x, Py\} \cap B$ is contained in C (aff refers to the affine hull operator). Figure 3 (a) is a visualization of this property. The red dot marks Py. The line going through the bottom left corner and Py leaves the polytope at Py and there exists no ball in C such that the line segment in this ball lies fully in C. This property is also not difficult to prove formally. We summarize the result in the following simple lemma.

Lemma 1 *Given a compact convex set $C \subset \mathcal{H}$ and a $y \in \mathcal{H}$ there exists a minimal face F that contains Py and for every $x \in C \setminus F$ and any ball B centered at Py with radius greater than zero we have that* $\operatorname{aff}\{x, Py\} \cap B \not\subset C$.

The existence of the minimal face is shown in Step (ii) of the proof of Theorem 2. If the second part of the lemma would be false then we would have a point in $\operatorname{aff}\{x, Py\} \cap C$ such that Py is a convex combination of x and this point. This would imply $x \in F$ (see eq. 3 on p. 28) with a contradiction to the assumptions of the lemma.

The lemma implies that any element outside the minimal face which is chosen by the algorithm will introduce perturbations into the estimate that can not be easily removed: the line from x through

Py leaves the convex set at Py and there is no element $z \in C$ that is well aligned with $-(x - Py)$, i.e. a component of $-(x - Py)$ points outward of the convex set and there exists no z in C such that $z - Py$ has a positive inner product with this component. These perturbations get only smaller over time because of the down-scaling of the old approximation at steps t by $t/(t+1)$ and not because of future choices of the algorithm which are well aligned with the error. We are able to show that despite this property the herding algorithm converges with the same rate of $1/t$ that it would achieve if Py would lie inside C under an assumption on these perturbations.

Three subspaces. To state our assumption, it is convenient to introduce the centered versions $C_c = \{x - Py : x \in C\}$ of C and $F_c = \{x - Py : x \in F\}$ of F. We denote the linear subspace spanned by C_c with $\operatorname{span} C_c$. We can write $\operatorname{span} C_c$ as a direct sum $U_F \oplus U_\delta \oplus U_b$ of three orthogonal subspaces $U_F, U_\delta, U_b \subseteq \operatorname{span} C_c$ and we denote the orthogonal projections onto these subspaces and onto $\operatorname{span} C_c$ by P_F, P_δ, P_b and P_C. Here,

- $U_F = \operatorname{span} F_c$ ($U_F = \{0\}$ if $F_c = \{0\}$) is the (unique) subspace spanned by the minimal face,
- U_δ and U_b are any two orthogonal subspaces of $\operatorname{span} C_c$ that are also orthogonal to U_F and which fulfill
  1. $\operatorname{span} C_c = U_F \oplus U_\delta \oplus U_b$,
  2. there exists a ball (relative to the subspace U_δ) around 0 in $\{P_\delta x : x \in C_c\}$,
  3. there exists an orthonormal basis of U_b, say $e_1, \ldots, e_k$, $k = \dim U_b$, such that for all $i \leq k$, $\{0\} \neq \{\langle e_i, x\rangle : x \in C_c\} \subset (-\infty, 0]$,
  4. $P_C(y - Py) \in U_b$.

  $U_b = \{0\}$ and $U_\delta = \{0\}$ are possible.

Observe that this representation is not invariant under rotations. In particular, U_δ and U_b can change dimensions when C is rotated (see Figure 4 (a) for an example). The split into these three subspaces is helpful since it allows us to separate the errors that can be minimized by choosing particular elements $x \in S$ from the errors that cannot be reduced by choosing elements in S. Recall that the errors of our algorithms at step t are w_t. The perturbations at step t that cannot be minimized by suitable choices of elements in S are $P_b w_t$ and the remainder $P_\delta w_t + P_F w_t$ consists of the error that can be reduced by choosing appropriate elements in S. The split into these subspaces is shown in Figure 3 (b). U_F is here the subspace spanned by the line on top of the triangular prism, U_δ is the subspace orthogonal to it and aligned with the base of the prism. U_b is orthogonal to these two subspaces and the whole prism lies below Py (the red dot) in U_b.

A perturbed optimization problem. Apart from adding to the overall error, these perturbations introduce a subtle and more serious problem as follows. For any $(P_F + P_\delta)w_t$ we have an element s^* that maximizes $\langle s^*, (P_F + P_\delta)w_t\rangle$ and by choosing s^* we reduce the overall error (the reduction will be significant if $\|(P_F + P_\delta)w_t\|$ is large). Imagine there is a second element s which has a very small positive inner product with $(P_F + P_\delta)w_t$, i.e. $0 < \epsilon_t = \langle s, (P_F + P_\delta)w_t\rangle \ll \langle s^*, (P_F + P_\delta)w_t\rangle$. s will not be chosen if the algorithm optimizes over $U_F \oplus U_\delta$, however, since we optimize over all of $\operatorname{span} C_c$, it is possible that $\langle s, w_t\rangle > \langle s^*, w_t\rangle$. In particular, this may happen if $\langle P_b w_t, s^*\rangle / \langle P_b w_t, s\rangle$ is large and the perturbations $P_b w_t$ are effectively changing the geometry

of the optimization problem in $U_F \oplus U_\delta$ (see Figure 3 (c) for a visualization of this effect). The rate of convergence can be significantly reduced by this effect. In particular this happens if the perturbations make the algorithm choose a sequence of elements with inner products ϵ_t converging to 0 since the error in $U_F \oplus U_\delta$ will then not decrease over time. This problem can only occur if the $P_b w_t$ values affect the different elements in S in a very unbalanced way, i.e. there must be some elements $x, x' \in S$ such that $\langle P_b w_t, x \rangle \gg \langle P_b w_t, x' \rangle$. Our theorem below shows that this is the central problem that can hinder the rate of convergence. We demonstrate that Algorithm 1 converges with a rate of $1/t$ if, and only if, the perturbations affect the different elements in S in a sufficiently balanced way.

The main assumption and the theorems. Assumption 1 below formalizes the concept of a balanced influence of the perturbations on the elements of S. We need the following set of *critical points* for a given sequence of elements x_t chosen by the algorithm to state this assumption,

$$D(\{x_t\}) = \{x : x \in S \backslash F, x_t = x \text{ for infinitely many } t\}.$$

We use here $\{x_t\}$ to abbreviate $\{x_t\}_{t \geq 1}$. Only elements in $D(\{x_t\})$ can lead to a reduction in convergence rates and it suffices to check that none of the elements in $D(\{x_t\})$ lead to the above described effect to demonstrate fast convergence. Our main assumption is now the following:

Assumption 1: *Given sequences $\{x_t\}_{t \geq 1}, \{w_t\}_{t \geq 0}$ we assume that there exists a representation $U_F \oplus U_\delta \oplus U_b = \operatorname{span} C_c$ as described above, where for all $x, x' \in D(\{x_t\})$ there exists a $\Delta < \infty$ and a $t' < \infty$ such that $\langle -P_b w_t, x - Py \rangle \leq \Delta \langle -P_b w_t, x' - Py \rangle$ for all $t \geq t'$.*

We restrict our analysis to the case where there are only finitely many extremes of C. We thus assume that C is compact, convex and has only finitely many extremes. A set C with these properties is called a *convex polytope*. The following theorem demonstrates that for a convex polytope the above assumption is both necessary and sufficient for the fast rate of convergence of Algorithm 1.

Theorem 2 *Given a compact convex set $C \subset \mathcal{H}$ and a finite subset S of C with $\operatorname{ex} C \subseteq S$ there exists for $y \in \mathcal{H}$ a constant $b > 0$ such that Algorithm 1 has a worst-case approximation error of b/t for all $t \geq 1$ if, and only if, Assumption 1 is fulfilled for the sequences $\{x_t\}_{t \geq 1}$ and $\{w_t\}_{t \geq 0}$ generated by the algorithm.*

The reason why the herding algorithm retains its rate of convergence is that elements outside the minimal face will not be chosen anymore after some time $t_0 \in \mathbb{N}$ and the decay $t/(t+1)$ is high enough to remove the perturbations introduced in these first t_0 many steps fast enough so that they do not harm the overall rate of convergence, i.e. if we assume $Py = 0$ then

$$\|(I - P_F) w_t\| = \frac{t-1}{t} \|(I - P_F) w_{t-1}\| = \frac{t-2}{t} \|(I - P_F) w_{t-2}\| = \frac{t_0}{t} \|(I - P_F) w_{t_0}\|$$

and the error orthogonal to the minimal face decays with a rate of $1/t$. The situation is different for the CGM. While we can show that no elements outside the minimal face is chosen after some $t_0 \in \mathbb{N}$ the perturbations introduced in these initial steps dominate the rate of convergence of the algorithm even for steps $t \gg t_0$ (in the case of applying the CGM with away-steps it is known that the minimal face is identified after finite many steps under certain conditions and the algorithms optimizes after

this step over the minimal face (Wolfe, 1970; Guélat and Marcotte, 1986)). We decompose the error in the theorem below into two parts, the error in the affine subspace spanned by minimal face F and the error in the orthogonal complement of that space. This decomposition is natural in the way that we have a strong reduction of the error in the minimal face per step and only a weak reduction in the orthogonal complement of the minimal face. The first part of the theorem is standard and follows from Beck and Teboulle (2004). In the following, p_t denotes the approximation at time t. We use again Assumption 1, but be aware the the weight w_t is defined slightly differently for the CGM, i.e. the weight is scaled down by a factor of $1/t$ at step t.

Theorem 3 *Given a compact convex set $C \subset \mathcal{H}$, a finite subset S of C with $\operatorname{ex} C \subseteq S$ and an element $y \in \mathcal{H}$ the following holds:*

- *If only elements in $F \cap S$, where $F \subset C$ is the minimal face that contains Py, are chosen then the method converges linearly to the projection and there exist constants $b, \beta > 0$ with*

$$\|Py - p_t\| \leq be^{-\beta t}.$$

- *Under Assumption 1 if $\min_{x \in S} \|Py - x\| > 0$ and the approximation does not equal Py in finite many steps then the sequence $\{\|(I - P_F)(Py - p_t)\|\}_{t \geq 0}$ converges sub-linearly and there exists a constant $d > 0$ such that*

$$\|P_F(Py - p_t)\|^2 \leq (1 - \delta_F^2/\operatorname{diam}^2(F)) \|P_F(Py - p_{t-1})\|^2 + d \|(I - P_F)(Py - p_{t-1})\|^2 .$$

  Furthermore, there exists a time t_0 after which only elements in $F \cap S$ are chosen.

It is worth noting that the term $(1 - \delta_F^2/\operatorname{diam}^2(F)) \|P_F(Py - p_{t-1})\|^2$ would guarantee a linear rate of convergence if the extra perturbation part would not be present.

The constant. It is of obvious interest to get insight into the size of the involved constants, in particular into the size of the constant b in Theorem 2 and its relation to quantities like δ_F and the dimension of $\operatorname{span} C_c$. The baseline is here the constant that we gain in the *trivial* case where $F = C$. Here, b can be taken to be $3r + 2r^2/\delta_F$, with $r = \sup_{x \in C} \|x\|$, and the constant does not depend on the affine dimension of C (see the proof of Theorem 2, part (ii.f)). In the case where $F \neq C$ we can first observe that b depends both on δ_m, the radius of the largest ball inside the projection of C_c onto $U_F \oplus U_\delta$, and on $\Delta' = \sup_{x,x' \in D(\{x_t\})} \Delta_{x,x'}$, where $\Delta_{x,x'}$ are the constants in Assumption 1, i.e. b depends on δ_m/Δ' (see the proof of Theorem 2, part (iv.b)). So the larger the ball around Py in $U_F \oplus U_\delta$ and the closer coupled the perturbations are the smaller the constant b. Referring these quantities back to geometric properties of C is non-trivial and in all likelihood providing a general characterization is at least as difficult as providing a general characterization of when Assumption 1 is fulfilled. However, in concrete settings it is often actually rather easy to provide bounds on b depending, for instance, on the affine dimension of C. We provide a number of examples in the next section. In particular, Corollary 6 - 8 contain concrete values for the constant. The dependence on d is here linear or sub-linear if we ignore the dependence of the dimension on the size of the set C, i.e. $r = \sup_{x \in C} \|x\|$ might also depend on the dimension. For instance, for the standard hypercube in d-dimensions $r = \sqrt{d}$.

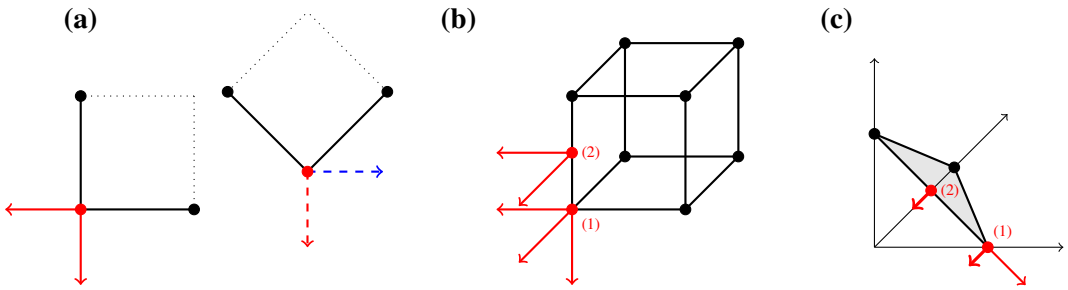

Figure 4: (a) The figure shows two possible splits into $U_b \oplus U_\delta$ depending on how the square is rotated. The first one is visualized by solid red arrows. Here $U_b = \mathbb{R}^2$ and $U_\delta = \{0\}$. The second split is visualized through the dashed arrows. The red dashed arrow is a basis vector of U_b and the blue dashed arrow is a basis vector of U_δ. (b) The figure shows two projection problems. The locations of Py are marked with the red dots and are annotated with (1) and (2). The red arrows visualize the space U_b for the two problems. For problem (1) this is a three dimensional space since U_F is here 0-dimensional and for (2) it is a 2 dimensional space. In both cases $U_\delta = \{0\}$. (c) The figure resambles (b). Instead of the hypercube it shows the simplex in three dimensions. The red arrows depict basis vectors for U_b. $U_\delta = \{0\}$ as for the hypercube.

2.1.1 EXAMPLES

We now take a look at a number of concrete projection problems to demonstrate how Assumption 1 can be used to prove fast convergence of Algorithm 1 in various situations. Our first result does not rely on assumptions about the shape of the convex polytope, but assumes that its span is a $d < \infty$ dimensional subspace of a Hilbert space and the projection Py lies in a $d-1$ dimensional face of the convex set (Corollary 4 and 5). In the applications we have in mind the point of projection is related to an estimator and the convex polytope enforces some form of sparsity. Typically, in such settings, the projection lies in some low dimensional face of the convex polytope.

We provide a rather general condition for this case in Corollary 6 and we use this corollary to demonstrate that Algorithm 1 converges with the fast rate independent of the location of Py if we are working, for instance, with the hypercube or the simplex (Corollary 7 and 8). In particular, Py can lie in a low dimensional face of the simplex or can be an extreme of it. There is certainly more to be understood here, but these cases demonstrate that the performance of Algorithm 1 is robust across a variety of settings and they demonstrate the uses of Assumption 1.

The proofs of the corollaries that follow are contained in the appendix on page 39 and onward. It seems also worth pointing out that we can rotate our coordinate system in an arbitrary way and translate the convex polytope and y without it affecting the algorithms or the rate of convergence. This holds because the algorithm uses only inner products and orthogonal operators (rotations) do not change these. Similarly, a translation does not affect the maximization step or the update. This allows us, for instance, to prove the fast rate of convergence for the standard hypercube $[0,1]^d$ and to generalize this result to arbitrary hypercubes in $\mathbb{R}^d$.

One-Dimensional U_b. If the space U_b is one-dimensional then the perturbations affect all elements in $S \backslash F$ equally (up to a finite multiplier) and Assumption 1 is always fulfilled. This is in particular the case if the minimal face is $(d-1)$-dimensional, but can also be fulfilled for lower dimensional faces if U_δ is not 0-dimensional.

Corollary 4 *Given a compact convex set $C \subset \mathcal{H}$ and a finite subset S of C with* $\operatorname{ex} C \subseteq S$. *If for $y \in \mathcal{H}$ there exists a decomposition $U_F \oplus U_\delta \oplus U_b$ of* $\operatorname{span} C_c$ *such that U_b is one dimensional then Assumption 1 is fulfilled. In particular, in this case there exists a constant $b > 0$ such that Algorithm 1 has a worst-case approximation error of b/t for all $t \geq 1$.*

One can relate the assumption that U_b is one-dimensional to a uniqueness assumption about the projection Py. If $P_C y \notin C$ and only elements in $A = \{z \in \mathcal{H} : z = \alpha P_C(y - Py) + P_C Py, \alpha \in [0, \infty)\}$ are projected onto Py then U_b is 1-dimensional and Assumption 1 is fulfilled. We summarize this result in another corollary.

Corollary 5 *Given a compact convex set $C \subset \mathcal{H}$ and a finite subset S of C with* $\operatorname{ex} C \subseteq S$. *Assumption 1 is fulfilled if $P_C y \in \mathcal{H} \backslash C$ and whenever $Pz = Py$ for some $z \in \mathcal{H}$ then $P_C z \in A$ holds. In particular, in this case there exists a constant $b > 0$ such that Algorithm 1 has a worst-case approximation error of b/t for all $t \geq 1$.*

This last assumption is not fulfilled in Figure 1 if the projection lies in the corners or edges, but it is fulfilled in all other cases. A quite different application of our theorem allows us to exploit the geometry of the convex set to derive the fast rate of convergence independent of the location of Py.

Zero-Dimensional U_δ. If we can rotate the convex set such that we get a split $U_F \oplus U_\delta \oplus U_b$ for which $U_\delta = \{0\}$ then our Assumption 1 is also fulfilled. The next corollary states this result. We use here, and in the following two corollaries, $d = \dim U_b$ and we let $e_1, \ldots, e_d$ be any basis of U_b. Furthermore, $\alpha = \min_{i \leq d} \min\{|\langle e_i, x - Py\rangle| : x \in S \backslash F, \langle e_i, x - Py\rangle \neq 0\} > 0$ and $r = \sup_{x \in C} \|x\|$.

Corollary 6 *Let C be a compact convex set in some Hilbert space $\mathcal{H}$, S a finite set with* $\operatorname{cch} S = C$, *and $y \in \mathcal{H}$ such that there exists a split into $U_F \oplus U_\delta \oplus U_b$ of* $\operatorname{span} C_c$ *with $U_\delta = \{0\}$. Assumption 1 is fulfilled and there exists a constant $b > 0$ such that Algorithm 1 has a worst-case approximation error of b/t for all $t \geq 1$. The constant b can be chosen as $\sqrt{d}\, 4r^3/(\alpha\delta_F) + 6r^2(1/\delta_F + 1/\alpha) + 5r$.*

This condition is somewhat abstract but leads, for example, to results that prove that the algorithm converges with the fast rate in classical sparsity settings independent of the location of Py (Corollary 7 and 8 below). Also, observe that Corollary 6 together with Corollary 4 imply that if Py lies in a face of dimension $(d-2)$ then Algorithm 1 will converge with the fast rate: if U_δ is 0-dimensional then this follows from Cor. 6 and if U_δ is 1-dimensional then U_F is also 1-dimensional and the result follows from Cor. 4 (U_δ cannot be 2-dimensional due to Lemma 1).

Hypercubes. Here, we consider the compact convex set $[0,1]^d = \operatorname{cch}\{0,1\}^d \subset \mathbb{R}^d$ and transformations of it (rotations, translations and changes of its size). The set of extremes of the standard hypercube is $\{0,1\}^d$ and the d-dimensional hypercube has $2^{d-m}\binom{d}{m}$ many m-dimensional faces, where $m \leq d$. We formulate the following corollary in terms of a rotation matrix Q, a translation by a vector z and a scaling by a scalar c. In the proof of the corollary we transform the hypercube to the standard hypercube and we show that $U_\delta = \{0\}$ independent of the location of y. This is visualized

in Figure 4 (b) for a 3-dimensional hypercube. (1) and (2) are here two projection problems and the red arrows depict two bases of U_b. Constructing such bases for which $U_\delta = \{0\}$ is always possible for the standard hypercube.

Corollary 7 *Let $y \in \mathbb{R}^d$, Q any orthogonal matrix, $c > 0$ any scaling, $z \in \mathbb{R}^d$, $S = \{0,1\}^d$ and $C = [0,1]^d$, $d \geq 1$. Assumption 1 is fulfilled for the set $\tilde{S} = cQ[S + z]$ and $\tilde{C} = cQ[C + z]$ (independently of the dimensionality of the face Py lies in). In particular, there exists a constant $b > 0$ such that Algorithm 1 has a worst-case approximation error of b/t for all $t \geq 1$. The constant b can be chosen as $\sqrt{d}\,4r^3/(c\delta_F) + 6r^2(1/\delta_F + 1/c) + 5r$.*

Standard Simplex. Algorithm 1 also attains a fast rate of convergence if the convex set we use is the standard simplex in $\mathbb{R}^d$ independently of the location of y. The standard simplex has $\binom{d}{m}$ many faces of dimension $m-1$, i.e. $1 \leq m \leq d$ and the dimension of the corresponding span of the centered face is $m-1$. The faces of such a simplex are again simplices. In particular, if we denote the standard simplex with $\Delta^{d-1} = \operatorname{cch}\{e_1, \ldots, e_d\}$, then the set of $m-1$ dimensional faces of Δ^{d-1} are $\{\operatorname{cch}\{e_{i_1}, \ldots, e_{i_m}\} : i_1 < i_2 \ldots < i_m, i_j \leq d\ \forall j \leq m\}$. As for the hypercube we can show that $U_\delta = \{0\}$. This is visualized in Figure 4 (c) for the standard simplex in $\mathbb{R}^3$. The figure shows two projection problems (1) and (2) and corresponding bases of U_b.

Corollary 8 *Let $y \in \mathbb{R}^d$, $S = \{e_i : i \leq d\}$ and $C = \Delta^{d-1} = \operatorname{cch} S$, $d \geq 1$. Assumption 1 is fulfilled and there exists a constant $b > 0$ such that Algorithm 1 has a worst-case approximation error of b/t for all $t \geq 1$. The constant b can be chosen as $4dr^3/\delta_F + 6r^2(1/\delta_F + \sqrt{d}) + 5r$.*

2.2 The Algorithm for Finite S

We are particularly interested in the case where $S = \{x_1, \ldots, x_n\}$ is finite. If S is finite we can change the algorithm to keep track of $\langle w_t, x_i\rangle =: a_{ti}$ instead of $w_t \in \mathcal{H}$. This reflects a change of representation from a basis representation of w_t to one based on S. The weight vector can, after this change of representation, be replaced by $a_t = (a_{t1}, \ldots, a_{tn})$ in the algorithm.

Algorithm 2. Input: $y \in \mathcal{H}$, $T \in \mathbb{N}$, and $S \subset \mathcal{H}$.

1. [Initialize] Set $a_0 = (\langle y, x_1\rangle, \ldots, \langle y, x_n\rangle)$ and $t = 1$.
2. [Optimization oracle] Choose $i_t \in \arg\max_{j \leq n} a_{(t-1)j}$.
3. [Update weight] Set $a_t = a_{t-1} - (\langle x_{i_t}, x_1\rangle, \ldots, \langle x_{i_t}, x_n\rangle) + a_0$.
4. [Iterate] If $t < T$ then increment t by 1 and go back to **2**.
5. [Terminate] Return the approximation $(x_{i_1} + \ldots + x_{i_T})/T$.

The computational costs per iteration are then determined by the number of samples n and the computational costs of calculating inner products in $\mathcal{H}$ (n inner products per iteration).

2.3 Line Search for Finite S

The line search in Algorithm 1 (ls) contains terms of the form $\|w_{t-1}\|^2$. The philosophy in Algorithm 2 is to avoid measuring objects in their norm and to work solely with relations between objects in the form of inner products $\langle y, x_i\rangle$ to reduce computational costs. To follow this philosophy we need a version of the line search that does not need access to $\|w_{t-1}\|$ or $\|y\|$. Achieving this goal is

easy enough: the line search aims for finding an α_t such that $p_t = (1-\alpha_t)p_{t-1} + \alpha_t x_{i_t}$, with p_t being our approximation at step t, achieves minimal distance to y. So we want that

$$\alpha_t \in \operatorname*{arg\,min}_{\alpha \in [0,1]} \|(1-\alpha)p_{t-1} + \alpha x_{i_t} - y\|^2 .$$

Setting the derivative of the squared norm to zero with respect to α we gain

$$\tilde{\alpha}_t = \frac{\langle y - p_{t-1}, x_{i_t} - p_{t-1}\rangle}{\|p_{t-1} - x_{i_t}\|^2}.$$

The norm in the denominator can be calculated recursively and calculating its value needs no significant computational resources. Because $\|w_t\| = \|p_t - y\|$ we gain the same $\tilde{\alpha}_t \in [0,\infty)$ as in Algorithm 1 (ls). Also, the minimizer over the interval $[0,1]$ is again $\alpha_t = 1 \wedge \tilde{\alpha}_t$.

2.4 Approximation Error Bounds through Duality

There exists a well established test for the approximation error (Wolfe, 1976). Let $S = \{x_1, \ldots, x_n\}$ still be finite. For any $p \in C$, $p \neq y$, we can write $\langle p, Py\rangle = \langle p, \alpha_1 x_1 + \ldots \alpha_n x_n\rangle \geq \min_{i \leq n} \langle p, x_i\rangle$ and

$$\|p - y\| \geq \|Py - y\| \geq \sum_{i=1}^{n} \alpha_i \big\langle x_i - y, \frac{p-y}{\|p-y\|}\big\rangle \geq \min_{x \in S} \big\langle x - y, \frac{p-y}{\|p-y\|}\big\rangle.$$

So

$$\|p - y\| - \|Py - y\| \leq \|p - y\| - \min_{x \in S} \big\langle x - y, \frac{p-y}{\|p-y\|}\big\rangle.$$

The last term tells us how far we are away from the optimum and it can be used to guarantee a specific approximation error at termination. In the settings we are interested in $\|p - y\|$ is, in fact, too expensive to compute and we need to modify the approach from Wolfe (1976). Observe that

$$\max_{x \in S} \big\langle p - x, \frac{p-y}{\|p-y\|}\big\rangle = \|p - y\| - \min_{x \in S} \big\langle x - y, \frac{p-y}{\|p-y\|}\big\rangle$$

gives us the easier to compute

$$\max_{x \in S} \big\langle p - x, p - y\big\rangle \geq \|p - y\| \left(\|p - y\| - \|Py - y\|\right)$$

The right side is directly an upper bound on $(\|p - y\| - \|Py - y\|)^2$, because $\|p - y\| \geq \|p - y\| - \|Py - y\| \geq 0$. However, this bound is overly conservative and we loose an order of magnitude since we lower bound $\|p - y\| \geq \|Py - y\|$ with a quantity that goes to 0. A better way seems to be to use linear functionals to approximate $p - y$. Natural choices are here $\langle x, \cdot\rangle$ for $x \in S$ or $\langle p, \cdot\rangle$ because we can often compute these functionals efficiently. The former choice can be exploited in the following way

$$\|y - p\| \geq \max_{x \in S} \big|\big\langle y - p, \frac{x}{\|x\|}\big\rangle\big|$$

and we can calculate a lower bound on $\|y - p\|$ in about $O(n)$ operations (depending on the representation of p) by calculating the right hand side inner product for every $x \in S$. Using this bound we get the following upper bound on the approximation error

$$\frac{\max_{x \in S} \langle p - x, p - y\rangle}{\max_{x \in S} |\langle y - p, x/\|x\|\rangle|} \geq \|p - y\| - \|Py - y\| .$$

The latter choice leads to

$$\|y - p\| \geq \langle y - p, \frac{p}{\|p\|} \rangle = \langle y, \frac{p}{\|p\|} \rangle - \|p\| .$$

The inequality also holds when taking the absolute value of the right side and

$$\max_{x \in S} \frac{\langle p - x, p - y \rangle}{|\|p\| - \langle y, p/\|p\| \rangle|} \geq \|p - y\| - \|Py - y\| .$$

If our convex set is a norm ball then as $p \to Py$ we also have that $y - p$ gets more and more aligned with $p/\|p\|$ and we expect this bound to be good.

2.5 Parallelization

Algorithm 2 is easy to parallelize since the bottleneck per iteration is the calculation of n inner products in the update equation of a_t. These inner products can be calculated independently of each other and by having c processes available we can distribute the computation such that each process has to calculate at most $\lceil n/c \rceil$ many inner products per iteration. Determining the arg max can then be achieved by a loop over the n entries of a_t. This operation is typically fast and no parallelization is needed. Though, it is easy to distribute this operation too to reduce the computation time to $\lceil n/c \rceil + c$ by first calculating c maxima over sets of size at most $\lceil n/c \rceil$ and by then calculating the maximum of these c maxima. Finally, the summation of the vector of inner products with $a_{t-1} - a_0$ can also be split such that each process has to perform at most $\lceil n/c \rceil$ many of these summations. This gives us in total a computation time in the order of $\lceil n/c \rceil$.

3. CCP-Kernel Regression

We now want to apply the algorithm to a challenging statistical problem. That is the problem of non-parametric regression. For this we use a reproducing kernel Hilbert space (RKHS, Aronszajn (1950)) which is in a certain sense natural given past herding applications, but it is also computationally efficient thanks to the reproducing property.

An RKHS $\mathcal{H}$ is implicitly defined through a kernel function $k(x, x')$. $\mathcal{H}$ is the completion of

$$L := \Big\{ \sum_{i=1}^{n} \alpha_i k(x_i, \cdot) : n \in \mathbb{N}, x_i \in X, \alpha_i \in \mathbb{R} \Big\},$$

where X is the domain of the covariates (or inputs). In statistical applications we like to approximate $\mathcal{H}$ with a nested family of sets of functions that are restricted in a certain way in their size. One nested family is, for instance, the family of closed balls B_r of radius r centered at zero. For these we know that $\bigcup_{r \geq 0} B_r = \mathcal{H}$. The balls B_r are, however, not compact if $\mathcal{H}$ is infinite dimensional. Also, controlling the extremes of B_r is not necessarily easy since we usually only have access to the kernel function k and not a basis $\{e_n\}_{n \geq 1}$ of $\mathcal{H}$. So ideally we would like a family of sets that can approximate $\mathcal{H}$ like the balls B_r, but in contrast to B_r the different sets would be compact, convex and controllable through the kernel. There is a family of sets with these properties which arises naturally from the kernel function (cl denotes the closure operator):

$$C(r) = \operatorname{cch} \{k(x, \cdot) : x \in X\} = \operatorname{cl} \Big\{ \sum_{i=1}^{n} \alpha_i k(x_i, \cdot) : n \in \mathbb{N}, x \in X^n, \alpha \in \mathbb{R}^n_+, \|\alpha\|_{\ell_1} \leq r \Big\} \subset \mathcal{H}.$$

$C(r)$ is compact and convex and the extremes of $C(r)$ are contained in the set $S = \{k(x,\cdot) : x \in X\}$. There is an obvious similarity to the definition of L and by suitably symmetrizing $C(r)$ we gain a family of sets which covers all of L and can approximate any element in $\mathcal{H}$ up to arbitrary precision.

3.1 Symmetric Closed-Convex Hull

The sets $C(r)$ might not contain 0 or any elements on the negative axes. This is obviously not ideal for representing functions. We overcome this shortcoming by symmetrising $C(r)$ by including the elements $-k(x,\cdot)$. In this way we get the family of sets

$$C_s(r) = \mathrm{cl}\left\{\sum_{i=1}^{n} \alpha_i k(x_i,\cdot) : n \in \mathbb{N}, x \in X^n, \alpha \in \mathbb{R}^n, \|\alpha\|_{\ell_1} \leq r\right\}.$$

This family of sets has some similarity to the closed balls in $\mathcal{H}$. In particular, if the kernel function is bounded then $C_s(r)$ is a subset of the closed ball of radius $\tilde{r} = r \sup_{x\in X} k(x,x)$ in $\mathcal{H}$ since

$$\left\|\sum_{i=1}^{n} \alpha_i k(x_i,\cdot)\right\| \leq \|\alpha\|_{\ell_1} \sup_{x\in X} k(x,x) \leq \tilde{r}$$

and the smallest closed set containing the elements $\sum_i \alpha_i k(x_i,\cdot)$ can not be larger than a closed ball of radius $\tilde{r}$.

Denseness and Universal Approximation. More important for us is the following property: the set $\bigcup_{r\geq 0} C_s(r)$ is *dense* in $\mathcal{H}$, since for any $f \in \mathcal{H}$ and $\epsilon > 0$ there exists an $n \in \mathbb{N}$, elements $x_1,\ldots,x_n \in X$ and coefficients $\alpha_1,\ldots,\alpha_n \in \mathbb{R}$ such that $\left\|f - \sum_{i=1}^n \alpha_i k(x_i,\cdot)\right\| \leq \epsilon$. But $\sum_{i=1}^n \alpha_i k(x_i,\cdot)$ is an element of $C_s(r)$ if $r \geq \sum_{i=1}^n |\alpha_i|$. In other words, we can approximate any function in $\mathcal{H}$ arbitrary well by making r large enough. Furthermore, if our kernel function is a universal kernel then we can approximate any continuous function on X arbitrary well in the supremums norm. This property is sometimes called the universal approximation property.

3.2 Simplifying the Sets

Optimizing over $C_s(r)$ itself is difficult and from a practical point of view it makes sense to reduce the sets further to save computation time. In regression we have a number of covariates $x_1,\ldots,x_n$ given and we know that we can represent any RKHS function h exactly on these points $x_i, i \leq n$, by functions of the form $\sum_{i=1}^n \alpha_i k(x_i,\cdot)$, where $\alpha \in \mathbb{R}^n$ are suitable weights, if the kernel matrix $K = (k(x_i,x_j))_{i,j\leq n}$ is of full rank because

$$K\alpha = \begin{pmatrix} h(x_1) \\ \vdots \\ h(x_n) \end{pmatrix}$$

has then a (unique) solution. From this point of view it makes sense to use the family of sets

$$\mathcal{C}(r) := \left\{\sum_{i=1}^{n} \alpha_i k(x_i,\cdot) : \alpha \in \mathbb{R}^n, \|\alpha\|_{\ell_1} \leq r\right\}.$$

instead of $C_s(r)$. $\mathcal{C}(r)$ is fully characterized by $S = \{\pm rk(x_1, \cdot), \ldots, \pm rk(x_n, \cdot)\}$ in the sense that $\mathcal{C}(r) = \operatorname{cch} S$ and S contains the extremes of $\mathcal{C}(r)$. S itself is of size $2n$ and we can optimize efficiently over it.

3.3 Interpolation in $\mathcal{H}$

Before approaching the regression problem we start with a closely related interpolation problem. We aim for interpolating a function $g \in \mathcal{H}$ at n support points $x_1, \ldots, x_n$. Let $y_1 = g(x_1), \ldots, y_n = g(x_n)$. The interpolation algorithm is essentially an efficient version of Algorithm 2 applied to the elements $\pm rk(x_1, \cdot), \ldots, \pm rk(x_n, \cdot)$, where r is the scaling factor of $\mathcal{C}(r)$. So at the heart of the algorithm we have a vector $a \in \mathbb{R}^n$ which keeps track of the approximation error and the entries a_i are just $\langle w_t, k(x_i, \cdot)\rangle$, where w_t is the weight vector used in Algorithm 1. We formulate the algorithm in terms of "$\leftarrow$" assignments which denote the operation of overwriting the left hand side with the value on the right hand side.

Algorithm 3. Input: $y \in \mathbb{R}^n$, $T \in \mathbb{N}$, $r > 0$, a kernel function k and $x_1, \ldots, x_n \in X$.

1. [Initialize] Set $a = ry$ and $t = 1$.

2. [Optimization oracle] Choose $j' \in \arg\min_{i \le n} a_i$, $j'' \in \arg\max_{i \le n} a_i$.

3. If $-\min_{i \le n} a_i \ge \max_{i \le n} a_i$ then let $j = j'$, $z_t = x_j$ and $s_t = 1$

4. else let $j = j''$, $z_t = x_j$ and $s_t = -1$.

5. [Update weight] Set $a_i \leftarrow a_i + ry_i - r^2 s_t k(x_i, z_t)$ for all $i \le n$.

6. [Iterate] If $t < T$ then increment t by 1 and go back to **2**.

7. [Terminate] Return the regressor $f(x) = r(s_1 k(z_1, x) + \ldots + s_T k(z_T, x))/T$.

The complexity of the algorithm is $O(Tn)$ since we have T iterations and we need to update in each iteration $a \in \mathbb{R}^n$. The bottleneck in this algorithm is the evaluation of the $k(z_i, x)$.

Line search. The line search version of this algorithm is based on Section 2.3. We update recursively the elements η, γ, π, where at step t we have that $\eta = \|p_{t-1}\|^2$, $\gamma = \langle p_{t-1}, y\rangle$ and $\pi = (\langle p_{t-1}, k(x_1, \cdot)\rangle, \ldots, \langle p_{t-1}, k(x_n, \cdot)\rangle)$, to keep the complexity of the algorithm at $O(Tn)$.

Algorithm 3 (ls). Input: $y \in \mathbb{R}^n$, $T \in \mathbb{N}$, $r > 0$, a kernel function k and $x_1, \ldots, x_n \in X$.

1. [Initialize] Set $a = ry, t = 1, \eta = 0, \gamma = 0$ and $\pi = (0, \ldots, 0)$.

2. [Optimization oracle] Choose $j' \in \arg\min_{i \le n} a_i$, $j'' \in \arg\max_{i \le n} a_i$.

3. If $-\min_{i \le n} a_i \ge \max_{i \le n} a_i$ then let $j = j'$, $z_t = x_j$ and $s_t = 1$

4. else let $j = j''$, $z_t = x_j$ and $s_t = -1$.

5. [Calculate step size] Let $v = r^2(k(x_1, z_t), \ldots, k(x_n, z_t))$ and

$$\tilde{\alpha}_t = \frac{r s_t y_j - r s_t \pi_j - \gamma + \eta}{\eta - 2 r s_t \pi_j + v_j}, \quad \text{if } t = 1 \text{ let } \alpha_t = 1 \text{ and otherwise let } \alpha_t = 1 \wedge \tilde{\alpha}_t.$$

6. [Update weight] $a \leftarrow (1 - \alpha_t)a + \alpha_t(ry - s_t v)$.

7. [Update line search variables] $\eta \leftarrow (1-\alpha_t)^2\eta + 2\alpha_t(1-\alpha_t)r s_t \pi_j + \alpha_t^2 v_j$, $\pi \leftarrow (1-\alpha_t)\pi + \alpha_t(s_t/r)v$ and $\gamma \leftarrow (1 - \alpha_t)\gamma + \alpha_t s_t r y_j$.

8. [Iterate] If $t < T$ then increment t by 1 and go back to **2**.

9. [Terminate] Return the regressor $f(x) = r(s_1\beta_1 k(z_1, x) + \ldots + s_T\beta_T k(z_T, x))/T$, with $\beta_i = \alpha_i \prod_{n=i+1}^{T}(1 - \alpha_n)$.

The expensive calculation is here the calculation of $v = (k(x_1, z_t), \ldots, k(x_n, z_t))$.

Approximation error stopping rule. We can also use an approximation error of below ϵ as the stopping criterion by using bounds from Section 2.4. The algorithm below achieves this in $O(Tn)$ for the bound that uses our approximation p_t to gauge the approximation error. The algorithm is very similar to the line search algorithm because we need to store similar quantities for calculating the bound as for the line search. We update recursively the elements η, γ, π, where at step t (before the update) we have that $\eta = \|p_{t-1}\|^2$, $\gamma = \langle p_{t-1}, y\rangle$,$\pi = (\langle p_{t-1}, k(x_1, \cdot)\rangle, \ldots, \langle p_{t-1}, k(x_n, \cdot)\rangle)$. The complexity of the algorithm is again $O(Tn)$. The version shown below is for the line search. By replacing α_t with $1/t$ one can gain a version for the standard algorithm. We state only the changes to Algorithm 3 (ls).

Algorithm 3 (ls, ae). *(replace 8. and 9. in Algorithm 3 (ls)).* Input: $y \in \mathbb{R}^n$, $\epsilon > 0$, $r > 0$, a kernel function k and $x_1, \ldots, x_n \in X$.

8. [Upper bound] Calculate the upper bound

$$b = \max_{i \le n} \frac{\eta - r\pi_i - \gamma + r y_i}{\left|\sqrt{\eta} - \gamma/\sqrt{\eta}\right|} \vee \max_{i \le n} \frac{\eta + r\pi_i - \gamma - r y_i}{\left|\sqrt{\eta} - \gamma/\sqrt{\eta}\right|}.$$

9. [Terminate] If $b \le \epsilon$ return the regressor $f(x) = r(s_1\beta_1 k(z_1, x) + \ldots + s_t\beta_t k(z_m, x))$ where $\beta_i = \alpha_i \prod_{n=i+1}^{t}(1 - \alpha_n)$. Otherwise, go back to **2**.

As for the other two algorithms above the computationally most demanding operation in this algorithm is the calculation of $(k(x_1, z_t), \ldots, k(x_n, z_t))$.

3.4 Norm Minimizing Regressor

The interpolation algorithms can also be applied to the regression problem. Let the observations be $(x_1, y_1), \ldots, (x_n, y_n)$ and let K be the kernel matrix. If the kernel matrix is of full rank then with $\alpha = K^{-1}y$, where $y = (y_1, \ldots, y_n)^\top$, the function $h(x) = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot) \in \mathcal{H}$ fulfills $(h(x_1), \ldots, h(x_n)) = K\alpha = y^\top$ and the interpolation algorithm applied to h will converge under

the usual conditions with a rate of $1/t$, that is $\|h - g_t\|^2 \leq b^2/t^2$ with $g_t = (1/t)\sum_{i=1}^{t} k(z_i, \cdot)$. There are two interesting observations here: first, the algorithm minimizes the distance to h implicitly without knowledge of h itself. In fact, determining h numerically is usually impossible due to ill-conditioning. Second, the algorithm minimizes the distance in the RKHS norm and not in a least-squares sense. Both, the RKHS norm and the least-squares criterion can be seen as distance measures that are in certain ways related. For instance, a norm-distance of zero between two functions implies that they have the same least-square error and, furthermore, the least-squares error of our approximation is bounded by

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - g_t(x_i))^2 = \frac{1}{n}\sum_{i=1}^{n}|\langle h - g_t, k(x_i, \cdot)\rangle|^2 \leq \left(\frac{1}{n}\sum_{i=1}^{n}k(x_i, x_i)\right)(\|h - Ph\|^2 + c/t^2)$$

where Ph denotes the projection of h onto $\mathcal{C}(r)$ the constant c is $(b^2/n)\sum_{i=1}^{n} k(x_i, x_i)$.

One can also formulate the norm minimization problem as a convex optimization problem

$$\min_{\alpha \in \mathbb{R}^n} \alpha^\top (K\alpha - 2y) \quad \text{s.t.} \sum_{j=1}^{n} |\alpha_j| = r.$$

and use convex programs to find the projection. This representation has the drawback that it makes explicit use of the $n \times n$ kernel matrix K and for large scale problems this formulation needs significant amounts of memory.

4. Experiments

We conducted a set of experiments to gauge the performance of the approach. The first set of experiments was constructed to demonstrate the behavior of the error bounds and to compare the optimization routine with some standard optimization procedures. The second set of experiments compared the regressor with well established regressors in a small scale setting. The advantage of the small scale setting is that we can compare to methods like the GP regressor. The final set of experiments focused on a large scale benchmark data set and compared our method to the Fast-KRR, which is the state-of-the-art regressor for large scale problems.

4.1 Experiment 1: Empirical Rate of Convergence

The first set of experiments were conducted to explore basic properties of the optimization approach: 'how does a typical regression curve look like in comparison to a GP regression curve?' 'How do the error bounds relate?' 'How does the optimization routine behave in comparison to a general purpose optimizer?' In our first experiment we generated 1000 data points from a Gaussian process (Gaussian covariance function) with normal distributed noise (the right plot in Figure 5). We fitted the maximum a posteriori estimator (MAP, known hyper-parameters) to it (red curve) and then did split the data into an 800 and 200 batch to run a cross validation loop over the hyper parameters (we also used a Gaussian covariance but with an unknown width parameter). We ran the CCP algorithm for 20 (yellow) and 100 (purple) iterations (without line search). The red bars show the points and weights of the solution when the algorithm is run for 100 iterations. The black bars show the solution found by the cvx Matlab toolbox for a precision of about 10^{-16}. The next experiment was about the bounds. We used again 1000 data points generated as above and a fixed hyper-parameter (no cv loop) to see how the bounds behave as the number of iterations increases. There

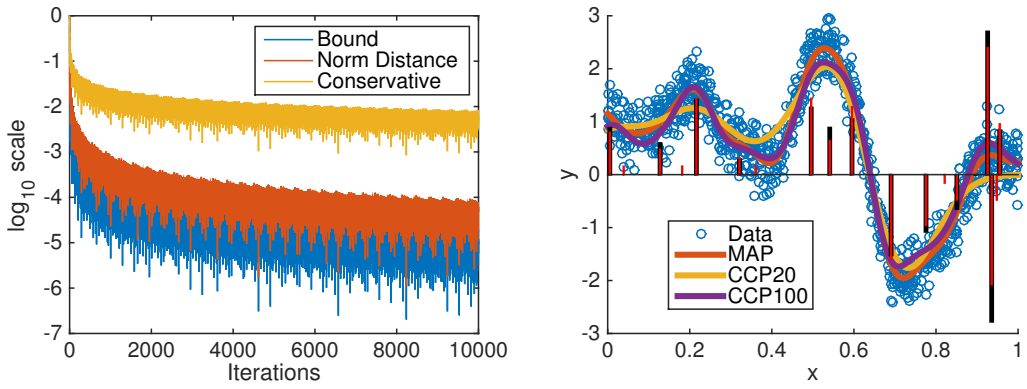

Figure 5: **Left:** The plot shows three different bounds on the approximation error. **Right:** The figure shows data from a Gaussian process with Gaussian likelihood function. The data is overlaid with three regression curves. MAP is here the optimal regressor. The vertical bars show the weight that is assigned by the CCP-regressors with 100 iterations (red) and the cvx matlab optimization toolbox (black) to observations at various locations x.

n	100	500	1000	2000	3000	4000	5000
CCP	0.03(0.03)	0.1(0.08)	0.2(0.09)	0.25(0.29)	0.3(0.26)	0.47(0.26)	0.57(0.61)
cvx	0.64(0.13)	1.3(0.08)	2.7(0.13)	8.16(0.12)	17.2(0.39)	31.3(1.68)	49.8(0.88)

Figure 6: Run time comparison between the CCP projection algorithm and a general purpose solver.

is one difficulty here, which is that we do not have the best fitting function $h \in \mathcal{H}$ and we can not calculate the exact distance to h. The three curves on the left side of the figure are our two bounds of the difference $\|h - g_t\| - \|h - Ph\|$, g_t is the regressor after t iterations and Ph the solution of the optimization problem which we try to find. *Conservative* refers to the bound where we use $\|g_t - h\| \geq (\|g_t - h\| - \|Ph - h\|)$ and *Bound* refers to our second bound) and the distance to Ph (*Norm Distance*), that is $\|g_t - Ph\|$ which is also an upper bound of $\|h - g_t\| - \|h - Ph\|$ (we determined Ph by using cvx with a precision of about 10^{-16}). The results are shown on the left in Figure 5. One can observe that all three bounds show similar behavior, however, the conservative bound is, as expected, very loose.

We were also interested in a run time comparison between the projection algorithm and a general purpose solver (the cvx toolbox) to see how much can be gained by using the specialized projection method. The cvx toolbox uses the SDPT3 package to solve semidefinite-quadratic-linear programming problems. The details of the SDPT3 package are described in Toh, Todd, and Tutuncu (1999); Tutuncu, Toh, and Todd (2003). We used as a stopping rule for both methods an error below 10^{-4}, that is CCP stopped when our bound guaranteed us that the error is below 10^{-4} and cvx stopped when its own error bound signaled an error below this threshold (details about how the precision is calculated in the SDPT3 package can be found in Toh et al. (1999)[Section 3]). The results of the run time comparison are shown in Figure 6 (mean(standard deviation) in seconds averaged over 10 runs; same experimental setup as for the bounds plot; n is the number of data points).

4.2 Experiment 2: Run time vs. Least-Squares Error on a Small Scale Sample

The second set of experiments tested on a small scale how the run time of the CCP-regressor measures against the statistical performance. We were interested in seeing how the method fares in comparison to standard Gaussian process/ridge regression and Fast-KRR. We reproduced the experiment from Zhang et al. (2013) which uses the million songs data set (about 450000 data points and a covariate dimension of 90). We normalized the data as in Zhang et al. (2013) by letting each covariate dimension have standard deviation 1. We also used the same kernel (Gaussian with $\sigma = 6$). Finally, we normalized the response variable (year when a song appeared) to lie in $[0, 1]$ by subtracting the minimal year and dividing by (maximum - minimum). 450000 data points are too much to run the standard GP regressor/ridge regressor and we downsampled the data set to a small subset of 5000 training points and 1000 test points. We were interested in how the radius affects the performance of CCP and how different it is from how λ affects the GP. We were also interested in seeing how the error threshold translates into least-squares performance and run time. The results are shown in the table in Figure 7. The notation is (run time - least-squares error), r is the radius for CCP (with line-search), p the number of elements in each partition of Fast-KRR (we used for Fast-KRR the same regularization schedule as in Zhang et al. (2013)) and ϵ, in CCP-ϵ, refers to the stopping criterion. We used our upper bound to gauge the current error and we stopped when the error bound passed ϵr. The table below shows the results. We also ran the CCP-0.1 setting with $r = 5000$ to see how close we can get to the performance of the GP regressor. The estimator took 30.6 seconds to produce an estimate with an error of 0.124.

$r(p)/\lambda$	50/0.02	100/0.01	250/0.004	500/0.002	1000/0.001
GP	56 - 0.121	55 - 0.121	55 - 0.121	55 - 0.121	65 - 0.121
CCP-0.1	1.03 - 0.53	1.63 - 0.46	2.97 - 0.36	4.77 - 0.22	8.89 - 0.17
CCP-0.01	3.66 - 0.47	6.21 - 0.33	12.8 - 0.19	30.67 - 0.16	93.13 - 0.13
Fast-KRR		9.3 - 0.236	10.6 - 0.203	12.5 - 0.179	16.5 - 0.158

Figure 7: The table shows a comparison between the GP/ridge regressor, the Fast-KRR and the CCP-regressor. Each entry consists of the run time (seconds) and the least-squares error.

As one expects the GP run time is essentially independent of λ. A bit surprising is here that the regularization parameter seems to have hardly any effect on the least-squares error of the GP regressor. The run time of the CCP algorithm, however, is strongly affected by r which is consistent with our bound in which r influences the constant of the convergence rate. For both thresholds the computation time increases slightly non-linearly in r. The least-squares error depends on both ϵ and r. Increasing r results in this experiment in a higher reduction in least-squares error in dependence of the added computation time.

The lowest error is achieved by the GP-regressor (which corresponds to the Fast-KRR with one partition). A slightly suboptimal solution with a least-squares error of 0.124 is achieved by the CCP-regressor in about 25 seconds less than what the GP-regressor needs. One can also observe that Fast-KRR is fast in computing an estimate with a low least-squares errors which is en par with

the CCP-regressor in this experiment (in 10 seconds the Fast-KRR reaches, for instance, about 0.2 and the CCP-regressor 0.17). The interesting question is here how the performance of Fast-KRR and the CCP-regressor scale with the amount of data.

4.3 Experiment 3: Run time vs. Least-Squares Error on a Large Scale Sample

In the last experiment we compared Fast-KRR with the CCP-regressor (line search) on the full million songs data set. We used similar partition sizes as in Zhang et al. (2013) for Fast-KRR and we ran the CCP-regressor with $r = 100000$. The bottleneck in the CCP-regressor is the number of kernel evaluations that need to be performed which are tn for t iterations and n samples. The Fast-KRR regressor needs for m partitions (for simplicity let m be such that $0 \equiv n \mod m$) in the order of n^2/m many kernel evaluations and it needs to calculate m many inverses of matrices of size $(n/m) \times (n/m)$. So if $t = n/m$ then the CCP-regressor needs to perform exactly as many kernel evaluations as the Fast-KRR algorithm. Especially for large m this will leave us with few iterations for the CCP-regressor and one expects that if the kernel evaluation is expensive in comparison to the computational cost of an inverse then Fast-KRR will perform better. On the other hand if we have either small partitions or cheap to evaluate kernels then the CCP-regressor will excel compared to Fast-KRR. In terms of memory: the CCP-regressor needs a small multiple of the original data size while the Fast-KRR needs to store another $n/m \times n/m$ matrix. We made 25 GB of memory available to the Fast-KRR method, which allowed us to go down to 26 partitions on our cluster. The results of the comparison are shown in Figure 8 (mean over 10 runs with randomly chosen training and test sets).

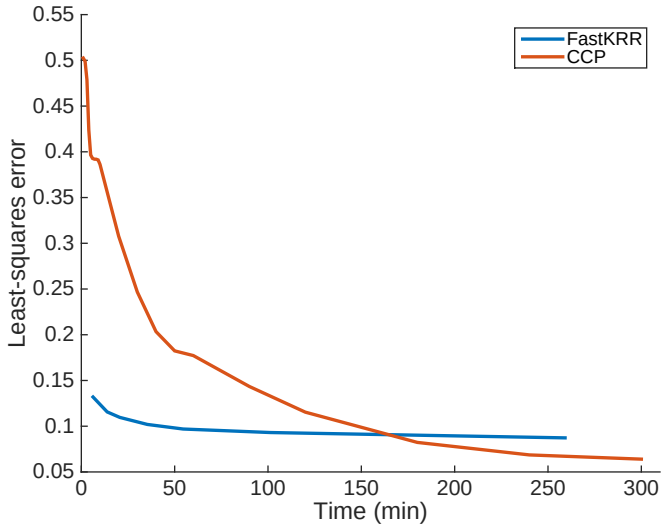

Figure 8: The plot shows the performance of the Fast-KRR and cpp-regressor in dependence of the run-time. We used partition sizes 26, 32, 38, 48, 64, 96, 128 and 256 for Fast-KRR. 26 was the limit we could achieve with 25 GB of memory.

We also determined the standard deviations. These were for both regressors marginal and we did not plot error-bars (maximal standard deviation for Fast-KRR: 0.16×10^{-4}; CCP: 0.02).

The plot shows us that the Fast-KRR achieves already a very good performance if the number of samples per partition is small. The CCP-regressor is significantly outperformed at that stage. With more run-time the CCP-regressor catches up and approaches the minimizer in the set $\mathcal{C}(100000)$. After about three hours of run-time the CCP-regressor overtakes the Fast-KRR and achieves the lowest least-squares error. Another difference one can observe is that there is, in principle, no limit of how long we can let the CCP-regressor optimize while we are limited with the Fast-KRR by the memory that we have available (the memory requirement increases linearly in the number of elements per partition and quadratically in the sample size).

5. Discussion and Open Problems

Motivated by the recent work on kernel herding we explored the uses of herding and the CGM algorithm for calculating projections onto convex sets. We derived a theorem that extends current convergence results for herding to the boundary if the perturbations introduced by our lack of knowledge of the minimal face that contains the projection are in a certain way well behaved. We also showed that our condition on the perturbations is both necessary and sufficient for the algorithm to converge with the fast rate. An important open question is if there exist convex sets for which this assumption can fail to hold. Furthermore, we demonstrated that the herding algorithm and the CGM will chose no elements outside of the minimal face after some finite time t_0 if this condition is fulfilled. Providing tight bounds on the size of t_0 and the number of elements outside the minimal face that are chosen in these first iterations in future research would help with improving the understanding of herding and the CGM further. Such bounds should also be useful to provide a better understanding of algorithms like the CGM with away steps: away steps are essentially trying to remove the perturbations introduced through elements outside of the minimal face. It is known that the CGM with away-steps successfully removes perturbations after finite many steps under certain conditions (e.g. Wolfe (1970); Guélat and Marcotte (1986)), though, as we discussed in the introduction, there is still much to be gained by refined analyses. Another interesting direction for future research concerns the geometry of the convex set in Hilbert space. The geometry factors into the rates of convergence of CGM like algorithms in various ways (Slater's condition, pyramidal width etc.). One might also aim for results that hold for convex sets with countably infinite or uncountable many extremes. We believe that one of the main difficulties will be here that there does not need to exist a gap between the minimal face and extremes that lie outside of the minimal face. We made use of such a gap to show that after some finite time only extremes of the minimal face will be chosen. Without such a gap one expects that there does not exist such a finite window of steps and perturbations will be continuously injected into the approximation.

On the practical side one can wonder how widely applicable the projection in kernel space approach, which we used to derive a novel kernel regressor, is. Projections onto the standard simplex have proven to be very valuable for a variety of statistical problems and the projection onto the set $\{\sum_{i=1}^{n} \alpha_i k(x_i, \cdot) : n \in \mathbb{N}, \|\alpha\|_{\ell_1} \leq 1, x_i \in X \text{ for } i \leq n\}$ is in a way the natural analogue of the standard simplex. There are some obvious differences. The set is, for example, from a geometrical point of view far more complicated since in kernel space the natural way to describe objects is through the kernel function and not through an orthonormal basis. Nevertheless, our results suggest that this approach has merit and it is worth to explore its uses for other non-parametric problems.

6. The Proofs of the Two Theorems and the Corollaries

This section contains the proof details of our two theorems and the various corollaries. We start with a technical lemma that is needed in the proof of the first theorem. Using this we prove that the basic algorithm without line-search achieves a convergence rate of $1/t$.

6.1 The First Theorem

We need a technical lemma that guarantees us that if we have two operators A, B that pull elements x towards the origin if x gets large (in a certain sense) then compositions of these operators (for instance $C = A^2BA^4B^3$) applied to x will be bounded, i.e. $\|Cx\| \leq b$ for some constant b. In fact, we need slightly more. We need a result which holds simultaneously for a family of operators $\mathfrak{A}$. We formulate the lemma in the way that some initial value x_0 and a particular infinite composition C of operators in $\mathfrak{A}$ and B is given. This is necessary since our operators in $\mathfrak{A}$ are not necessarily a contraction for arbitrary elements but only for the elements they are applied to. In the following, we denote with C_t the composition operator that consists of the first t operators in C (in the above example $C_3 = A^2B$) and we use the notation $C_{s,t}$ for the operator that fulfills $C_t = C_{s,t} \circ C_{s-1}$. We split the proof of the lemma and the following theorems into a number of separate claims and we use **P** (proof) and **Q** (q.e.d.) brackets for the proofs of the claims.

Lemma 9 *Let $\mathcal{H}$ be a Hilbert space, K a closed subspace of $\mathcal{H}$, P the projection onto K, $B : \mathcal{H} \to \mathcal{H}$, $Bx = (I - P)x + PBPx$ an operator, $\mathfrak{A}$ a family of operators $A : \mathcal{H} \to \mathcal{H}$, C a finite composition of operators of $\mathfrak{A}$ and B, and $x_0 \in \mathcal{H}$ such that there exist constants $a, b, \xi > 0$ with*

(i) $\|Ax\|, \|Bx\| \leq \|x\| + b$ *for every* $A \in \mathfrak{A}$, $x \in \mathcal{H}$.

(ii) For any element $x \in \{C_t x_0\}_{t \geq 1}$ and for any $t \geq 1$ for which $C_{t,t} \in \mathfrak{A}$, i.e. an operator in $\mathfrak{A}$ is chosen at time t, we have with $A = C_{t,t} \in \mathfrak{A}$ that

$$\|Ax\|^2 \leq \|x\| (\|x\| - \xi) + b.$$

(iii) If $\|Px\| \geq a$ for an element $x \in \mathcal{H}$ then $\|Bx\| \leq \|x\|$.

Then for all $t \geq 1$

$$\|C_t x_0\|^2 \leq \|x\|^2 \vee (c + b)^2 + (a + b)^2 \quad \text{and} \quad \|C_t x_0\| \leq \|x\| \vee (c + b) + a + b,$$

with the constant $c := ((a + b)^2 + b)/\xi$.

Proof **(a)** For any $n \geq 0$, $x \in \mathcal{H}$, it holds that $\|B^n x\|^2 \leq \|x\|^2 + (a + b)^2$.

P Observe that $Bx = (I-P)x + BPx$. $\|Px\| < a$ implies $\|BPx\| \leq a+b$ and if $\|Px\| \geq a$ then

$$\|BPx\|^2 = \|Bx\|^2 - \|(I - P)x\|^2 \leq \|x\|^2 - \|(I - P)x\|^2 = \|Px\|^2$$

and $\|BPx\| \leq \|Px\| \vee (a + b)$. Hence,

$$\|B^n Px\| \leq \left\|PB^{n-1}x\right\| \vee (a + b) = \left\|B^{n-1}Px\right\| \vee (a + b) \leq \|Px\| \vee (a + b).$$

So for any $n \geq 0$ and $x \in \mathcal{H}$ we know that

$$\|B^n x\|^2 = \|(I - P)x\|^2 + \|B^n Px\|^2 \leq \|(I - P)x\|^2 + \|Px\|^2 \vee (a+b)^2 \leq \|x\|^2 + (a+b)^2. \quad \textbf{Q}$$

(b) For any $n \geq 0$, $A \in \mathfrak{A}$, if $\|x\| \geq c + b$, $x = C_t x_0$ and $C_{t+1,t+n+1} = AB^n$ for some $t \geq 0$ then $\|AB^n x\| \leq \|x\|$ and $\|AB^n x\| \leq \|x\| \vee (c + b)$.

P If $\|x'\| \geq c$ then (ii) tells us that

$$\left\|Ax'\right\|^2 \leq \left\|x'\right\|^2 - \xi \left\|x'\right\| + b \leq \left\|x'\right\|^2 - (a+b)^2.$$

So, if $\|B^n x\| \geq c$ then (a) tells us that $\|AB^n x\|^2 \leq \|B^n x\|^2 - (a+b)^2 \leq \|x\|^2$. And, if $\|B^n x\| < c$ then $\|AB^n x\| \leq c + b$. Also, $\|AB^n x\| \leq \|x\|$ if $\|x\| \geq c + b$. **Q**

(c) For any $n \geq 0, m \geq 1$, $A_1, \ldots, A_m \in \mathfrak{A}$, if $\|x\| \geq c + b$, $x = C_t x_0$, $C_{t+1,t+n+m} = A_1 \ldots A_m B^n$ for some $t \geq 1$ then $\|A_1 \ldots A_m B^n x\| \leq \|x\|$ and it holds that $\|A_1 \ldots A_m B^n x\| \leq \|x\| \vee (c + b)$.

P If $\|z\| \geq b/\xi$ for a $z \in \mathcal{H}$ then $\|Az\| \leq \|z\|$ for all $A \in \mathfrak{A}$. If $\|x\| \geq c + b$ then (b) tells us that $\|AB^n x\| \leq \|x\|$ for all $A \in \mathfrak{A}$. Hence, for any $2 \leq l \leq m - 1$, $\|A_{m-l-1} \ldots A_m B^n x\| > \|A_{m-l} \ldots A_m B^n x\|$ implies $\|A_{m-l} \ldots A_m B^n x\| < b/\xi \leq c$. The maximal increase of operators $A \in \mathfrak{A}$ is bounded by b due to (i). Since we can only see an increase if $\|A_{m-l} \ldots A_m B^n x\| \leq c$ we gain $\|A_{m-l-1} \ldots A_m B^n x\| \leq (c + b) \vee \|x\| = \|x\|$. A slight modification of the argument yields the second case. **Q**

(d) It follows that compositions of such sequences, say $A_1 \ldots A_{m_1} B^{n_1} A_{m_1+1} \ldots A_{m_1+m_2} B^{n_2}$, $m_1, m_2 > 0$, $A_1, \ldots, A_{m_1+m_2} \in \mathfrak{A}$ do not increase the bound since with $x' = A_{m_1+1} \ldots A_{m_2} B^{n_2} x$,

$$\left\|x'\right\| = \|A_{m_1+1} \ldots A_{m_2} B^{n_2} x\| \leq \|x\| \vee (c + b)$$

we have

$$\left\|A_1 \ldots A_{m_1} B^{n_1} x'\right\| \leq \left\|x'\right\| \vee (c + b) \leq \|x\| \vee (c + b).$$

These cases cover all possible compositions with the only exception of sequences that start with some B^n, $n \geq 1$. But, if $\|x'\| \leq \|x\| \vee (c + b)$ then from (a) we know that

$$\|Cx\|^2 = \left\|B^n x'\right\|^2 \leq \left\|x'\right\|^2 + (a+b)^2 \leq \|x\|^2 \vee (c+b)^2 + (a+b)^2.$$

■

We need the definitions of the affine hull and the affine dimension. The *affine hull* of a set $A \subseteq \mathcal{H}$ is

$$\operatorname{aff} A = \left\{ \sum_{i=1}^{n} \alpha_i x_i : n \geq 1, x_i \in A, \sum_{i=1}^{n} \alpha_i = 1 \right\}.$$

The affine hull can be identified with a vector space by centering it around an arbitrary element $x_0 \in \operatorname{aff} A$, i.e. $V = \{x - x_0 : x \in \operatorname{aff} A\}$ is a vector space. The dimension of this vector space is called the *affine dimension* of A. If $A = \{x\}$ for some element $x \in \mathcal{H}$ then aff A is also just $\{x\}$ and we define its affine dimension to be 0.

We recall some basic properties of the projection Py of y onto a compact convex set C. The projection Py is characterized by $\|y - Py\| = \min_{x \in C} \|y - x\|$ and the geometric property $\langle y - Py, x - Py \rangle \leq 0$ for every $x \in C$. The latter property translates into a sort of orthogonal decomposition for convex sets, that is

$$\|y - x\|^2 = \|y - Py\|^2 - 2\langle y - Py, x - Py \rangle + \|Py - x\|^2 \geq \|y - Py\|^2 + \|Py - x\|^2. \quad (1)$$

We also need the definition of a face of a convex set C. A *face* F is a set which fulfills that whenever there are two points $a, b \in C$ and $\theta \in (0, 1)$ with $\theta a + (1 - \theta) b \in F$ then $a, b \in F$.

Theorem 2 *Given a compact convex set $C \subset \mathcal{H}$ and a finite subset S of C with $\operatorname{ex} C \subseteq S$ there exists for $y \in \mathcal{H}$ a constant $b > 0$ such that Algorithm 1 has a worst-case approximation error of b/t for all $t \geq 1$ if, and only if, Assumption 1 is fulfilled for the sequences $\{x_t\}_{t\geq 1}$ and $\{w_t\}_{t\geq 0}$ generated by the algorithm.*

Proof We start with some general observations before proving that the assumption is sufficient for the fast rate of convergence. That the condition is necessary is shown at the end (see (v)).

(i) The algorithm converges with the right rate to Py if the sequence $\{w_t - y - t(y - Py)\}_{t\geq 1}$ stays in a bounded norm ball of radius b since then

$$b \geq \|w_t - y - t(y - Py)\| = \left\|(t+1)y - \sum_{i=1}^{t} x_i - y - t(y - Py)\right\| = \left\|tPy - \sum_{i=1}^{t} x_i\right\| \quad (2)$$

because $w_t = (t+1)y - (x_1 + \ldots + x_t)$ and hence $\|Py - (x_1 + \ldots + x_t)/t\| \leq b/t$. Also observe that we can replace for any $z \in \mathcal{H}$

$$\arg\max_{x\in S} \langle z, x\rangle \quad \text{with} \quad \arg\max_{x\in S} \langle z, x - Py\rangle$$

in the maximization step.

(ii) (a) There exists a minimal face F that contains Py which is

$$F = \{Py\} \cup \{x \in C : \exists z \in C, \theta \in (0,1) \text{ such that } \theta x + (1-\theta)z = Py\}. \quad (3)$$

(b) F is a compact and convex set and the extremes of F are $\operatorname{ex} F = \operatorname{ex} C \cap F$.

Let $F_c = \{z - Py : z \in F\}$ be the face F centered around $Py \in F$. The set $\operatorname{aff} F_c = \operatorname{span} F_c$ is a closed subspace of $\mathcal{H}$ with orthogonal complement $(\operatorname{aff} F_c)^\perp = F_c^\perp$. So to any element $z \in \mathcal{H}$ we have a unique $z^{||} \in \operatorname{aff} F_c$ and $z^\perp \in (\operatorname{aff} F_c)^\perp$ with $z = z^{||} + z^\perp$. In the following, denote the projection onto $\operatorname{aff} F_c$ with P_F.

(c) $y - Py$ stands orthogonal on the centered face F_c, i.e. $y - Py \in F_c^\perp$. So the term $t(y - Py)$ contained in w_t does not influence the maximization over F.

(d) Either $F_c = \{0\}$ or there exists a $\delta_F > 0$ such that $B(0,\delta_F) \cap \operatorname{span} F_c = B(0,\delta_F) \cap F_c$, where $B(0,\delta_F)$ is the closed ball of radius δ_F centered at the origin.

(e) For any w_t, $t \geq 0$, there exists an $x \in C$ such that $\langle x - Py, w_t\rangle \geq 0$. Furthermore, if $F_c \neq \{0\}$ and whenever $\|P_F w_t\| \geq 2r^2/\delta_F$, $r := \sup_{x\in C} \|x\| < \infty$, and an element in $S \cap F$ is chosen by the algorithm then $\|P_F w_{t+1}\| \leq \|P_F w_t\|$. So, if $\|P_F w_t\| \geq 2r^2/\delta_F$ and an element in $S \cap F$ is chosen then $\|w_{t+1} - y - (t+1)(y - Py)\| \leq \|w_t - y - t(y - Py)\|$.

(f) If $F = C$ then $b = 2r^2/\delta_F + 3r$ satisfies eq. 2 and the error of the algorithm is bounded by $(2r^2/\delta_F + 3r)/t$ for any $t \geq 1$.

P (a) *F is the minimal face.* F is certainly a subset of any face that contains Py by the very definition of a face. It is also a face itself since if for any $x \in F$, $x \neq Py$, there exist two points $a, b \in C$ and a $\lambda \in (0,1)$ such that $x = \lambda a + (1-\lambda)b$ then there exists a $\theta \in (0,1)$ and a $z \in C$ such that $Py = \theta x + (1-\theta)z = \theta(\lambda a + (1-\lambda)b) + (1-\theta)z = \theta\lambda a + \theta(1-\lambda)b + (1-\theta)z = \theta\lambda a + (1-\theta\lambda)\left(\frac{\theta(1-\lambda)}{1-\theta\lambda}b + \frac{1-\theta}{1-\theta\lambda}z\right)$. $c := \theta(1-\lambda)/(1-\theta\lambda)b + (1-\theta)/(1-\theta\lambda)z$ is an element of C since it is a convex combination of b and z and hence with $\xi := \theta\lambda \in (0,1)$ we see that $Py = \xi a + (1-\xi)c$ and $a \in F$. Applying the same argument to b shows us also that $b \in F$ and F is a face.

(b) *F is compact and convex and* $\operatorname{ex} F = \operatorname{ex} C \cap F$. F is a face and as such also convex. Furthermore, $F = C \cap \operatorname{aff} F$ where $\operatorname{aff} F$ is the affine hull of F. The affine hull has finite affine dimension since C has finite affine dimension. Affine hulls with finite affine dimension are closed. Hence, F is closed as the intersection of two closed sets. And since closed subsets of compact sets are compact we see that F is compact. Finally, $\operatorname{ex} F = F \cap \operatorname{ex} C$, that is the extremes of F are just the extremes of C which lie in F. The last property can be verified in the following way: the extremes of C are by definition not convex combinations of any two points $a, b \in C$, $a \neq b$, and hence of no two points in F. So $F \cap \operatorname{ex} C \subseteq \operatorname{ex} F$. On the other hand, if $c \in \operatorname{ex} F$ then there is no point $a \in C \setminus F$ for which a corresponding point $b \in C$ and $\theta \in]0,1[$ exists such that $c = \theta a + (1-\theta)b$ (a would otherwise be in the face F). However, since $c \in \operatorname{ex} F$ there exists also no point $a \in F$ with this property and $c \in \operatorname{ex} C$.

(c) *$y - Py$ is orthogonal to F_c.* If there would be an element $u \in \operatorname{aff} F_c$, $u = \sum_{i=1}^{n} \alpha_i v_i$ for some $n \geq 1$, $v_i \in F_c$, $\sum_{i=1}^{n} \alpha_i = 1$, such that $\langle u, y - Py \rangle \neq 0$ there would also be an element $v \in F_c$ with $\langle v, y - Py \rangle \neq 0$ (otherwise $\langle u, y - Py \rangle = \sum_{i=1}^{n} \alpha_i \langle v_i, y - Py \rangle = 0$). Furthermore, by the characterization of the minimal face in (3) and because $v + Py \in F$ there exists an element $z \in F_c$ such that $Py = \theta(v + Py) + (1-\theta)(z + Py)$ and $0 = \theta v + (1-\theta)z$. Hence, $0 = \langle y - Py, \theta v \rangle + \langle y - Py, (1-\theta)z \rangle$. Because $\langle y - Py, v \rangle \neq 0$ we know that $\langle y - Py, z \rangle \neq 0$ and both have opposing signs. So one of them, say without loss of generality $\langle y - Py, v \rangle$, is strictly greater zero. But this can not be since then $\tilde{v} = v + Py \in C$ would fulfill, in contradiction to the properties of Py, $\langle y - Py, \tilde{v} - Py \rangle > 0$.

(d) *$F_c = \{0\}$ or there exists a $B(0,\delta), \delta > 0$.* If the minimal face consists of a single element x then this element must be Py, since $Py \in F$, so in this case $F_c = \{0\}$. In the other case it suffices to consider a set of basis vectors which span the space $\operatorname{span} F_c$, say $e_1, \ldots, e_d$, where $d < \infty$ is the affine dimension of F. There exist $\alpha_1, \ldots, \alpha_d > 0$ such that $\pm\alpha_i e_i \in F_c$ for all $i \leq d$ if, and only if, $B(0,\delta) \cap \operatorname{span} F_c = B(0,\delta) \cap F_c$ for some $\delta > 0$.

Since, $e_i \in \operatorname{aff} F_c$ there exists a $n \geq 1$, $u_1, \ldots, u_n \in F_c$ and $\sum_{j=1}^{n} \beta_j = 1$ with $e_i = \sum_{j=1}^{n} \beta_j u_j$. We can also represent e_i as a sum with only non-negative coefficients, i.e. there exists $\tilde{\beta}_j \geq 0, \tilde{u}_j \in F_c$, $j \leq n$, such that $e_i = \sum_{j=1}^{n} \tilde{\beta}_j \tilde{u}_j$. This can be achieved by doing the following for each $j \leq n$: if $\beta_j \geq 0$, then let $\tilde{\beta}_j = \beta_j$ and $\tilde{u}_j = u_j$. If $\beta_j < 0$ and $u_j = 0$ then let $\tilde{\beta}_j = -\beta_j$ and $\tilde{u}_j = u_j$. Finally, if neither case applies ($\beta_j < 0$, $u_j \neq 0$) take a $\theta \in]0,1[$ and a $\tilde{u}_j \in F_c$ such that $0 = \theta u_j + (1-\theta)\tilde{u}_j$. Such a θ and $\tilde{u}_j$ exist due to the definition of F. With this choice $\beta_j u_j = -\tilde{u}_j \beta_j (1-\theta)/\theta = \tilde{\beta}_j \tilde{u}_j$, where $\tilde{\beta}_j = -\beta_j(1-\theta)/\theta > 0$.

So $e_i = \sum_{j=1}^{n} \tilde{\beta}_j \tilde{u}_j$ and $\tilde{\beta}_j \geq 0$. By normalizing the sum with $\xi = \sum_{j=1}^{n} \tilde{\beta}_j > 0$ (some β_j must be greater 0) we see that $\xi e_i \in F_c$. Furthermore, there exists a $\theta \in]0,1[$ and a $v \in F_c$ such that $-e_i \xi(1-\theta)/\theta = v$. And our claim is shown to be true by letting $\alpha_i = \min\{\xi(1-\theta)/\theta, \xi\}$.

(e) *Shrinking weight vector.* If F consists of more than one element then for any weight vector w_t there exists an element x in F with $\langle x - Py, P_F w_t \rangle = \delta_F \|P_F w_t\|$, since $\delta_F P_F w_t / \|P_F w_t\|$ is in $B(0,\delta_F)$ and $\langle (\delta_F P_F w_t / \|P_F w_t\| + Py) - Py, P_F w_t \rangle = \delta_F \|P_F w_t\|$. This implies directly that we have an element $z \in \operatorname{ex} F \subseteq S \cap F$ with $\langle z - Py, P_F w_t \rangle \geq \delta_F \|P_F w_t\|$, because $x = \sum_{j=1}^{n} \beta_j u_j$, for suitable n, $\beta_1, \ldots, \beta_n > 0$, $\sum_{j=1}^{n} \beta_j = 1$ and elements $u_j \in \operatorname{ex} F$ ($\operatorname{ex} C$ is finite and the convex hull C of $\operatorname{ex} C$ is in this case closed. That means any element in C can be represented exactly by a convex combination of finite many extremes), i.e. $\sum_{j=1}^{n} \beta_j \langle u_j - Py, P_F w_t \rangle = \delta \|P_F w_t\|$ and at least one term $\langle u_j - Py, P_F w_t \rangle$ must be as large as $\delta_F \|P_F w_t\|$.

Now, whenever $\|P_F w_t\| \geq 2r^2/\delta_F$ and an element in $S \cap F$ is chosen, then $\|P_F w_{t+1}\| \leq \|P_F w_t\|$. This can be verified in the following way: there exists an element $x' \in F \cap S$ such that $\langle x' - Py, P_F w_t\rangle \geq \delta_F \|P_F w_t\|$ and, hence, the algorithm will choose an element x with $\langle x - Py, P_F w_t\rangle \geq \langle x' - Py, P_F w_t\rangle \geq \delta_F \|P_F w_t\|$ (as said in (i) the maximization is unaffected by the translation $-Py$). So, since $x - Py \in F_c$ and P_F is linear we have that

$$\begin{aligned}\|P_F w_{t+1}\|^2 &= \|P_F(w_t - (x - Py))\|^2 = \|P_F w_t - (x - Py)\|^2 \\ &= \|P_F w_t\|^2 - 2\langle x - Py, P_F w_t\rangle + \|x - Py\|^2 .\end{aligned}$$

Furthermore, $\|x - Py\| \leq \|x\| + \|Py\|$ is bounded by $2r$ and

$$\|x - Py\|^2 - 2\langle x - Py, P_F w_t\rangle \leq 4r^2 - 2\delta_F \|P_F w_t\| \leq 0$$

by our assumption on $\|P_F w_t\|$.

For $x \in F \cap S$ we can also observe that for any w_t

$$\langle x - Py, w_t\rangle = \langle P_F(x - Py), w_t\rangle = \langle x - Py, P_F w_t\rangle$$

since $x - Py \in \operatorname{aff} F_c$ and P_F is self-adjoint. So there always exists an $x \in C$ such that $\langle x - Py, w_t\rangle \geq 0$. If $F_c = \{0\}$, that is $F = \{Py\}$, then $x = Py$ gives us $\langle x - Py, w_t\rangle = 0$.

(f) If $F = \{Py\}$ then $w_t - y - t(y - Py) = (x_1 - Py) + \ldots + (x_t - Py) = 0$ since all $x_i = Py$. If F contains more than one element and $C = F$ then (e) tells us that if $\|P_F w_t\| \geq 2r^2/\delta_F$ then

$$\|w_{t+1} - y - (t+1)(y - Py)\| \leq \|w_t - y - t(y - Py)\| .$$

Observe that if $F = C$ then the projection P_F is closely related to the projection $\tilde{P}$ onto the affine hull of C, that is $\tilde{P}x = P_F(x - \tilde{P}0) + \tilde{P}0$. Also, $\tilde{P}y = Py$ because $y - Py$ is orthogonal to span F_c. We need to show that $\|P_F w_t\| \geq 2r^2/\delta_F$ under the stated condition. We have that

$$\|P_F w_t\| = \Big\|(t+1)P_F y - \sum_{i=1}^{t} P_F x_i\Big\| = \Big\|P_F y - \sum_{i=1}^{t} P_F(x_i - y)\Big\| = \Big\|P_F y - \sum_{i=1}^{t}(x_i - Py)\Big\|$$

because $x_i - Py$ lies in span F_c and $P_F y = P_F Py$. On the other hand

$$\|w_t - y - t(y - Py)\| = \Big\|\sum_{i=1}^{t}(x_i - Py)\Big\| \leq \|P_F w_t\| + \|P_F Py\| \leq \|P_F w_t\| + r.$$

Hence, if $\|w_t - y - t(y - Py)\| \geq r + 2r^2/\delta_F$ then $\|w_{t+1} - y - (t+1)(y - Py)\|$ is smaller than $\|w_t - y - t(y - Py)\|$. So we can only see an increase of the sequence $\{w_t - y - t(y - Py)\}_{t\geq 1}$ if at any given t it holds that $\|w_t - y - t(y - Py)\| < 2r^2/\delta_F + r$ and since the change between t and $t+1$ is just $Py - x$ we gain for any $t \geq 1$ the bound

$$\|w_t - y - t(y - Py)\| < 2r^2/\delta_F + r + \|Py - x\| \leq 3r + 2r^2/\delta_F . \qquad \blacksquare$$

(iii) The case $F = C$ is dealt with in (ii.f) so we may assume in the following that $C \setminus F \neq \emptyset$.

We assign to the three subspaces U_F, U_b and U_δ defined in Section 2.1 orthonormal bases. Take an arbitrary orthonormal basis of U_F and let E_F be the set of these basis elements. Similarly, chose

a basis of U_δ and let E_δ be the set of these basis elements. Finally, group the basis elements of U_b, guaranteed to us in our assumption, in E_b. Furthermore, let δ_m be the largest radius of an open ball around 0 in $U_F \oplus U_\delta$. δ_m is always strictly larger than 0. For any $e \in E_b$ and $t \geq 0$

$$\langle w_t, e\rangle = \langle Py, e\rangle + \langle (t+1)(y-Py), e\rangle - \sum_{n=1}^{t} \langle x_n - Py, e\rangle \geq \langle (t+1)(y-Py), e\rangle + \langle Py, e\rangle$$

and for any $e \in E_b$ and any $t \geq 0$ it holds that

$$\langle w_{t+1} - (y-Py), e\rangle - \langle w_t, e\rangle = -\langle x_{t+1} - Py, e\rangle \geq 0.$$

Hence $\langle w_{t+1} - (y-Py), e\rangle \geq \langle w_t, e\rangle$ for all $e \in E_b$.

Clearly, F_c lies in $U_F \oplus U_\delta$. More importantly, any element that is not in F_c lies at least partly in U_b, or in other words, if $x \in C\backslash F$ then $P_b(x-Py) \neq 0$.

P For any element $x \in C$ that is not in F there exists an $e \in E_b$ such that $\langle x - Py, e\rangle \neq 0$. Otherwise, $x - Py$ would lie in a subspace A for which there is a $\delta' > 0$ and $B(0,\delta') \cap A = B(0,\delta') \cap A \cap C_c$. The element $-(x-Py)$ would then also lie in A and $z = -(\delta'/2)(x - Py)/\|x-Py\| \in B(0,\delta') \cap A = B(0,\delta') \cap A \cap C_c$ ($x \neq Py$, since $Py \in F$ and the norm $\|x-Py\|$ is strictly positive). So with $\zeta = \delta'/(2\|x-Py\|)$ and $\xi = \zeta/(1+\zeta) \in]0,1[$ we would have that

$$\xi(x-Py) + (1-\xi)z = (x-Py)\zeta/(1+\zeta) - (x-Py)\zeta(1-\zeta/(1+\zeta)) = 0$$

and $\xi x + (1-\xi)(z+Py) = Py$. But, because x and $z + Py \in C$ this implies that $x \in F$ by the definition of the minimal face. The conclusion $x \in F$ is a contradiction to our original assumption and our claim holds. **Q**

(iv) (a) If the sequence $\{\|(P_F + P_\delta)(w_t - y - t(y-Py))\|\}_{t\geq 1}$ is bounded then the sequence $\{\|P_b(w_t - y - t(y-Py)\|\}_{t\geq 1}$ is also bounded and elements in $S\backslash F$ are chosen only finitely often.

(b) Under Assumption 1 the sequence $\{\|(P_F + P_\delta)(w_t - y - t(y-Py))\|\}_{t\geq 1}$ is bounded. Furthermore, under this assumption the sequence $\{\|w_t - y - t(y-Py)\|\}_{t\geq 1}$ is bounded and there exists a constant b such that the approximation error is bounded by b/t.

P (a) Let us assume that $\{\|P_b(w_t - y - t(y-Py)\|\}_{t\geq 1}$ is unbounded. If elements in F are chosen at any stage t then $P_b(w_t - y - t(y-Py)) = P_b(w_{t+1} - y - (t+1)(y-Py))$ and there is no increase of the normed sequence. Hence, there must exist an element $x^* \in S\backslash F$ which is selected infinitely often. Let $e = -P_b(x^* - Py)/\|P_b(x^* - Py)\|$ and observe that $\langle e, x - Py\rangle \leq 0$ for all $x \in C$ because $\langle P_b(x^* - Py), x - Py\rangle = \sum_{e' \in E_b} \langle e', x^* - Py\rangle \langle e', x - Py\rangle \geq 0$. If x^* is selected at any step t then from $\langle x^* - Py, w_t\rangle \geq 0$ it follows that

$$\langle x^* - Py, (P_F + P_\delta)w_t\rangle \geq -\langle x^* - Py, P_b w_t\rangle = \|P_b(x^* - Py)\| \langle e, w_t\rangle$$

and $\langle x^* - Py, y - Py\rangle \leq 0$ implies $\langle x^* - Py, (P_F + P_\delta)(y - Py)\rangle \leq -\langle x^* - Py, P_b(y-Py)\rangle$ which in turn implies

$$\begin{aligned}\langle x^* - Py, (P_F + P_\delta)(-y - t(y-Py))\rangle &\geq -\langle x^* - Py, P_b(-y - t(y-Py))\rangle - \langle x^* - Py, Py\rangle \\ &\geq -\langle P_b(x^* - Py), P_b(-y - t(y-Py))\rangle - 2r.\end{aligned}$$

Together, these inequalities give us

$$\left\langle x^* - Py, (P_F + P_\delta)(w_t - y - t(y - Py))\right\rangle \geq \left\| P_b(x^* - Py)\right\| \langle e, P_b(w_t - y - t(y - Py)\rangle - 2r.$$

Applying the Cauchy-Schwarz inequality yields then

$$\frac{\|x^* - Py\|}{\|P_b(x^* - Py)\|} \|(P_F + P_\delta)(w_t - y - t(y - Py))\| + \frac{2r}{\|P_b(x^* - Py)\|} \geq \langle e, w_t - y - t(y - Py)\rangle$$
$$\geq -\sum_{s=1}^{t} \langle e, x_s - Py\rangle \times \chi\{x_s = x^*\} = \sum_{s=1}^{t} \left\| P_b(x^* - Py)\right\| \times \chi\{x_s = x^*\}$$

where χ is the characteristic function. Since $\|P_b(x^* - Py)\| > 0$ and x^* is selected infinitely often we conclude that $\|(P_F + P_\delta)(w_t - y - t(y - Py))\|$ diverges in t and the corresponding sequence $\{\|(P_F + P_\delta)(w_t - y - t(y - Py)\|\}_{t\geq 1}$ is unbounded.

(b) If $U_F = U_\delta = \{0\}$ then $(P_F + P_\delta)x = 0$ for all $x \in \mathcal{H}$ and (a) gives us

$$\|w_t - y - t(y - Py)\| = \left\|\sum_{i=1}^{t}(x_i - Py)\right\| = \left\|\sum_{i=1}^{t} P_b(x_i - Py)\right\| = \|P_b(w_t - y - t(y - Py)\|$$

and the sequence is bounded due to (a). Hence, the result follows.

Now let us assume that $U_F \oplus U_\delta \neq \{0\}$. We want to apply Lemma 9. Let $\tilde{\mathcal{H}} := U_F \oplus U_\delta$. $\tilde{\mathcal{H}}$ together with the inner product inherited from $\mathcal{H}$ is a Hilbert space. Let A be the operator defined for all $w \in \mathcal{H}$ through

$$x' \in \operatorname*{arg\,max}_{x \in S\setminus F} \langle x - Py, w\rangle \quad \text{and} \quad Aw = w + y - x'.$$

For any w for which there exist multiple maximizer assign to Aw one of these maximizer. Let B be defined in the same way with the only difference that $S\setminus F$ is replaced by $S \cap F$. We like to study the interaction between A and B on the subspace $U_F \oplus U_\delta$. The operator B optimizes over elements in $S \cap F$. For such elements $x \in S \cap F$ we know that $x - Py$ is orthogonal to $P_b w$ for all possible weights w and $P_b w$ does not influence the behavior of the operator B. Let $\tilde{B}$ be $(P_F + P_\delta)B$ restricted to $\tilde{\mathcal{H}}$ and let $K := U_F \subseteq \tilde{H}$. The operator $\tilde{B}$ fulfills the assumptions of the lemma:

- $\tilde{B}$ leaves the space orthogonal to K in $\tilde{H}$ unchanged and the maximization over $S \cap F$ does only depend on $P_F w$ for any $w \in \tilde{\mathcal{H}}$ so $\tilde{B}w = (I - P_F)w + P_F \tilde{B} P_F w$. This also holds (trivially) if $F_c = \{0\}$.

- Let $b := 2r$ then for $w \in \tilde{\mathcal{H}}$, $\left\|\tilde{B}w\right\| = \|(P_F + P_\delta)(w + (y - Py) - (x - Py))\| \leq \|w\| + \|x - Py\| \leq \|w\| + b$ where $x \in S \cap F$ and $(P_F + P_\delta)(y - Py) = 0$ by our choice of U_δ.

- If $F_c = \{0\}$ then point (iii) in Lemma 9 holds trivially. In all other cases let $a := 2r^2/\delta_F$, $\delta_F > 0$, then an argument as in (ii.e) tells us that if $\|P_F w\| \geq a$ for some $w \in \tilde{\mathcal{H}}$ then $\left\|\tilde{B}w\right\| \leq \|w\|$.

The operator A depends on $P_b w$ and we need to account for this error when studying the behavior of A acting on $U_F \oplus U_\delta$. We do so by working with a family of operators A which are parameterized by $P_b w$. Let

$$\mathfrak{A} = \{\tilde{A}_v : \tilde{\mathcal{H}} \to \tilde{\mathcal{H}} : \tilde{A}_v(w) = (P_F + P_\delta)A(w+v) \text{ for all } w \in \tilde{\mathcal{H}}, v \in U_b, \langle v, e\rangle \geq 0, \forall e \in E_b\}.$$

Observe that there is some time $t'' < \infty$ such that for all $t \geq t''$ no element in $S\backslash(F \cup D(\{x_t\})$ is chosen and each element in $D(\{x_t\})$ has been chosen at least once. Write $U_b = U \oplus U'$ with $U' = \text{span } D(\{x_t\})$ and a suitable subspace U. Then $P_U w_t = P_U w_{\tilde{t}}$ for all $t \geq \tilde{t}$. Since each element in $D(\{x_t\})$ is chosen infinitely often and $\langle -P_b w_t, x - Py\rangle$ is non-decreasing for $x \in C$ there exists some element $t''' \geq t''$ such that for all $t \geq t'''$, $\langle P_U(x - Py), -P_b w_t\rangle \leq \langle x' - Py, -P_b w_t\rangle$ for all $x \in S\backslash D(\{x_t\})$ and $x' \in D(\{x_t\})$. Let $\tilde{t} < \infty$ be the maximum over all t' in Assumption 1 and t'''. The family of operators $\mathfrak{A}$ fulfills the assumptions of the lemma if Assumption 1 holds and if we use $x_{\tilde{t}}$ as the initial value.

- Given any $w \in \tilde{\mathcal{H}}$, $\tilde{A} \in \mathfrak{A}$ (with corresponding $v \in U_b, \langle v, e\rangle \geq 0$ for all $e \in E_b$), there exists a $x \in S\backslash F$ such that

$$\begin{aligned}\left\|\tilde{A}w\right\| &= \|(P_F + P_\delta)A(w+v)\| = \|(P_F + P_\delta)(w + v + (y - Py) - (x - Py))\| \\ &= \|w - (x - Py)\| \leq \|w\| + b.\end{aligned}$$

- If $D(\{x_t\}) = \emptyset$ then only elements in F are chosen after $\tilde{t}$ and the claim of the theorem follows by the arguments in (ii.e).

  In case that $D(\{x_t\}) \neq \emptyset$ consider the maximum over all Δ in Assumption 1 and call this maximum $\tilde{\Delta} < \infty$. We claim that there exists a constant $\Delta' < \infty$ such that with $\xi := 2\delta_m/\Delta'$ for all $w_t, t \geq \tilde{t}$, whenever $x_t \in S\backslash F$, then

$$\langle x_t - Py, (P_F + P_\delta)w_t\rangle \geq \|(P_F + P_\delta)w_t\|\,\delta_m/\Delta'$$

  where x_t is the element selected in the argmax step of A_v, $w_t = u + v, u \in U_F \oplus U_\delta, v \in U_b$. **P** ($\alpha$) Whenever $x_t \in S\backslash F, x_t \in D(\{x_t\})$, for some $t \geq \tilde{t}$ we know that $\langle P_b w_t, x_t - Py\rangle \neq 0$ because each element $x \in D(\{x_t\})$ has been chosen at least once and each element outside the minimal face lies partly in U_b. For any $x \in D(\{x_t\})$ we also have $\langle P_b w_t, x - Py\rangle \neq 0$ and at step $t \geq \tilde{t}$

$$\begin{aligned}&\langle x_t - Py, (P_\delta + P_F)w_t\rangle \geq \langle x - Py, (P_F + P_\delta)w_t\rangle - \langle (x - Py) - (x_t - Py), -P_b w_t\rangle \\ &\geq \langle x - Py, (P_F + P_\delta)w_t\rangle - \left(\frac{\langle x - Py, -P_b w_t\rangle}{\langle x_t - Py, -P_b w_t\rangle} - 1\right)\langle x_t - Py, -P_b w_t\rangle \\ &\geq \langle x - Py, (P_F + P_\delta)w_t\rangle - \left(\frac{\langle x - Py, -P_b w_t\rangle}{\langle x_t - Py, -P_b w_t\rangle} - 1\right)\langle x_t - Py, (P_F + P_\delta)w_t\rangle \\ &\geq \langle x - Py, (P_F + P_\delta)w_t\rangle - (\tilde{\Delta} - 1)\langle x_t - Py, (P_F + P_\delta)w_t\rangle\end{aligned}$$

  and

$$\langle x_t - Py, (P_\delta + P_F)w_t\rangle \geq \langle x - Py, (P_\delta + P_F)w_t\rangle / \tilde{\Delta}.$$

  (β) Consider a set of elements $z_1, \ldots, z_{\tilde{d}} \in D(\{x_t\})$ such that $z_1 - Py, \ldots z_{\tilde{d}} - Py$ are linearly independent and with $\dim U' = \tilde{d} \geq 1$. Apply the Gram-Schmidt method to transform these into an orthonormal basis $\tilde{z}_1, \ldots, \tilde{z}_{\tilde{d}}$ such that $\tilde{z}_i = \sum_{j=1}^{i} \beta_{ij}(z_j - Py)$ for some

scalars β_{ij}. Since the vectors are linearly independent we know that $\max_{i,j\le d}\left|\beta_{ij}\right| < \infty$. For any $x \in S\backslash(F \cup D(\{x_t\}))$ if $P_b x \in U$ then, by the choice of $\tilde{t}$, for all $t \ge \tilde{t}$

$$\langle -P_b w_t, x - Py\rangle \le \min_{x' \in D(\{x_t\})} \langle -P_b w_t, x' - Py\rangle.$$

If $P_b x \notin U$ then we can write $P_{U'}(x-Py) = \alpha_1 \tilde{z}_1, \ldots, \alpha_{\tilde{d}} \tilde{z}_{\tilde{d}}$ for suitable scalars $\alpha_1, \ldots, \alpha_{\tilde{d}}$. We claim that there exists a $\Delta_x < \infty$ such that

$$\langle -P_b w_t, P_{U'}(x - Py)\rangle \le \Delta_x \max_{i\le d} \langle -P_b w_t, z_i - Py\rangle$$

for all $t \ge \tilde{t}$. This holds since

$$\begin{aligned}\langle -P_b w_t, P_{U'}(x - Py)\rangle &\le d \max_{i\le d} |\alpha_i| \max_{i\le d} \langle -P_b w_t, \tilde{z}_i\rangle \\ &\le d^2 \max_{i\le d} |\alpha_i| \max_{i,j\le d} \left|\beta_{ij}\right| \max_{i\le d} \langle -P_b w_t, z_i - Py\rangle\end{aligned}$$

and, using Parseval's identity, $|\alpha_i| \le \|P_{U'}(x - Py)\| \le 2r$. Hence, the claim follows with $\Delta_x = 2rd^2 \max_{i,j\le d}\left|\beta_{ij}\right| < \infty$.

Furthermore, because for all $t \ge \tilde{t}$, $x \in S\backslash(F \cup D(\{x_t\}))$, $\langle P_U(x - Py), -P_b w_t\rangle \le \min_{i\le d}\langle z_i - Py, -P_b w_t\rangle$ by our choice of $\tilde{t}$ we can observe that for any $x \in S\backslash(F \cup D(\{x_t\}) =: M$

$$\begin{aligned}\langle -P_b w_t, x - Py\rangle &= \langle -P_b w_t, P_U(x - Py)\rangle + \langle -P_b w_t, P_{U'}(x - Py)\rangle \\ &\le (\sup_{x\in M} \Delta_x + 1) \max_{i\le d} \langle -P_b w_t, z_i - Py\rangle \\ &\le \tilde{\Delta}(\sup_{x\in M} \Delta_x + 1)\langle -P_b w_t, x_t - Py\rangle.\end{aligned}$$

Repeating the argument in (α) this tells us that

$$\langle x_t - Py, (P_\delta + P_F) w_t\rangle \ge \langle x - Py, (P_\delta + P_F) w_t\rangle / (\tilde{\Delta}(\sup_{x\in M} \Delta_x + 1)).$$

(γ) The above argument also works for $x \in F$ without the need to adapt the multiplier $\Delta' := \tilde{\Delta}(\sup_{x\in M} \Delta_x + 1)$. Now, we know that there exists an element $x^* \in S$ such that

$$\langle x^* - Py, (P_F + P_\delta) w_t\rangle \ge \|(P_F + P_\delta) w_t\| \, \delta_m$$

and, hence,

$$\langle x_t - Py, (P_F + P_\delta) w_t\rangle \ge \langle x^* - Py, (P_F + P_\delta) w_t\rangle / \Delta' \ge \|(P_F + P_\delta) w_t\| \, \delta_m / \Delta'$$

which proves the claim. ❶

Writing $u = w + v$, $w \in U_b \oplus U_\delta$, $v \in U_b$, this becomes $\langle x_t - Py, w\rangle \ge \|w\| \, \delta_m / \Delta'$ for all $t \ge \tilde{t}$. The usual argument gives us for the operator $\tilde{A}$ being used at time t that

$$\begin{aligned}\left\|\tilde{A} w\right\|^2 &= \left\|(P_F + P_\delta)(w + v + (y - Py) - (x' - Py))\right\|^2 \\ &= \|w\|^2 - 2\left\langle x' - Py, w\right\rangle + \left\|x' - Py\right\|^2 \\ &\le \|w\|^2 - 2\|w\| \, \delta_m / \Delta' + b = \|w\| \, (\|w\| - 2\xi) + b.\end{aligned}$$

Hence, the lemma can be applied and the sequence of weights projected onto $U_F \oplus U_\delta$, i.e. $\{(P_F + P_\delta)w_t\}_{t\geq 0}$ is bounded in norm. Together with (a) this implies that the weight sequence stays bounded and the result follows. ❶

(v) The condition is also necessary for the fast rate of convergence. Observe that there always exists a decomposition $U_F \oplus U_\delta \oplus U_b$ of $\operatorname{span} C_c$: let $U_F = \operatorname{span} F_c$. Let $\tilde{U}$ be the orthogonal complement of U_F in $\operatorname{span} C_c$. If this is empty then we are done. Otherwise, chose an orthonormal basis $e_1, \ldots, e_k$ of $\tilde{U}, k = \dim \tilde{U}$ such that $e_1 = P_C(y - Py)/\|P_C(y - Py)\|$ if $P_C(y - Py) \neq 0$. Consider the set $E_b = \{e_i : i \leq k, P_{e_i}[C_c] \subset (-\infty, 0] \text{ or } P_{e_i}[C_c] \subset [0, \infty)\}$ and let $E_\delta = \{e_1, \ldots e_k\} \backslash E_b$. Then $U_b = \operatorname{span} E_b, U_\delta = \operatorname{span} E_\delta$ fulfill the assumptions. Since $P_\delta[C_c]$ is convex and we have an open interval around 0 in each direction $e \in E_\delta$ we have a ball around 0 in U_δ. Similarly, the assumption for U_b is fulfilled if we exchange $e \in E_b$ with $-e$ whenever $P_e[C_c] \subset [0, \infty)$.

If Assumption 1 is not fulfilled for this decomposition then there exists an element $x \in S \backslash F$ which is selected infinitely often by the algorithm. But, since $\|P_b(w_t - y - t(y - Py))\|$ is non-decreasing in t and because $x \in S \backslash F$ fulfills $\|P_b(x - Py)\| > 0$, we have $\|w_t - y - t(y - Py)\| \geq \|P_b(x - Py)\| \sum_{s=1}^{t} \chi\{x_s = x\}$ and the right side diverges in t. Therefore, we have an unbounded sequence $\{\|w_t - y - t(y - Py)\|\}_{t\geq 1}$. Observe that the algorithm does not converge with the rate $1/t$ if $\{\|w_t - y - t(y - Py)\|\}_{t\geq 1}$ is unbounded. ■

6.2 The Second Theorem

Theorem 3 *Given a compact convex set $C \subset \mathcal{H}$, a finite subset S of C with $\operatorname{ex} C \subseteq S$ and an element $y \in \mathcal{H}$ the following holds:*

- *If only elements in $F \cap S$, where $F \subset C$ is the minimal face that contains Py, are chosen then the method converges linearly to the projection and there exist constants $b, \beta > 0$ with*

$$\|Py - p_t\| \leq be^{-\beta t}.$$

- *Under Assumption 1 if $\min_{x\in S} \|Py - x\| > 0$ and the approximation does not equal Py in finite many steps then the sequence $\{\|(I - P_F)(Py - p_t)\|\}_{t\geq 0}$ converges sub-linearly and there exists a constant $d > 0$ such that*

$$\|P_F(Py - p_t)\|^2 \leq (1 - \delta_F^2/\operatorname{diam}^2(F)) \|P_F(Py - p_{t-1})\|^2 + d \|(I - P_F)(Py - p_{t-1})\|^2 .$$

  Furthermore, there exists a time t_0 after which only elements in $F \cap S$ are chosen.

Proof We aim for a similar argument as in the proof of the first theorem: (1) optimizing the approximation of Py by using y instead of Py does not hurt the rate of convergence. (2) This guarantees us that the rate of convergence is what we aim for if we work solely with the minimal face F that contains Py. Since, we do not know this minimal face we need to deal with perturbations that are introduced by elements in $S \backslash F$. These perturbations do slow down the rate of convergence. We adapt the proof from Beck and Teboulle (2004) to address (1). We make use of parts of the proof of Theorem 2 and we use Theorem 2 (ii.a) etc. to refer to it. In the following we will use p_t to denote the approximation at step t (with $p_0 = 0$), x_t as the element that is chosen by the algorithm and $w_t = y - p_t$.

(i) Let F be the minimal face which contains Py (Thm. 2 (ii.a)) and assume that either $C = F$ or only elements in F are chosen by the algorithm. (a) We claim that in this case $\tilde{\alpha}_t \in [0,1]$ for all $t \geq 1$. **P** In step 1 we have by definition that $\tilde{\alpha}_t = 1$. If C consists of a single element then $\tilde{\alpha}_t = 0/0$, which we define to be 0, for all $t \geq 2$. For the case that C consists of more than a single element and for any step $t \geq 2$ we have that

$$\tilde{\alpha}_t = \frac{\langle w_{t-1}, w_{t-1} + (x_t - y)\rangle}{\|w_{t-1} + x_t - y\|^2} = \frac{\langle y - p_{t-1}, x_t - p_{t-1}\rangle}{\|x_t - p_{t-1}\|^2} = \frac{\langle Py - p_{t-1}, x_t - p_{t-1}\rangle}{\|x_t - p_{t-1}\|^2}$$

where the last step holds because $x_t - p_{t-1}$ lies in the span of C_c and $y - Py$ stands orthogonal on $\operatorname{span} C_c$ (Thm. 2 (ii.c)), i.e. $\langle y - Py, x_t - p_{t-1}\rangle = 0$. We can also observe that

$$\langle Py - p_{t-1}, x_t - Py\rangle = \max_{x \in C} \langle Py - p_{t-1}, x - Py\rangle \geq \delta_F \|Py - p_{t-1}\| \geq 0 \tag{4}$$

holds: $y - Py$ stands orthogonal on $\operatorname{span} C_c$ and, hence,

$$\arg\max_{x \in S} \langle w_{t-1}, x\rangle = \arg\max_{x \in S} \langle Py - p_{t-1}, x\rangle \subseteq \arg\max_{x \in C} \langle Py - p_{t-1}, x\rangle .$$

Furthermore, $Py - p_{t-1}$ lies in the span of the centered minimal face, i.e. $Py - p_{t-1} \in F_c$, and Thm. 2 (ii.d) tells us that there exists a constant $\delta_F > 0$ which makes the above true. Following Beck and Teboulle (2004) we complete the square

$$\begin{aligned}\langle Py - p_{t-1}, Py - p_{t-1} - (Py - x_t)\rangle &\leq \|Py - p_{t-1}\|^2 - 2\langle Py - p_{t-1}, Py - x_t\rangle + \|Py - x_t\|^2 \\ &= \|p_{t-1} - x_t\|^2\end{aligned}$$

and observe that $\tilde{\alpha}_t \leq 1$. Hence, $\alpha_t = \tilde{\alpha}_t$. **Q**

(b) If $F = C$ or only elements in F are chosen by the algorithm then for any $t \geq 2$ either $Py = p_t$ and for all $s \geq t$ we have $Py = p_s$ or $\|Py - p_t\|^2 \leq (1 - \delta_F^2/\operatorname{diam}^2(C))\,\|\tilde{w}_{t-1}\|^2$. Furthermore, there exist constants $b, \beta > 0$ such that $\|Py - p_t\| \leq be^{-\beta t}$. **P** We aim at reproducing the argument from Beck and Teboulle (2004) by exploiting the orthogonality between $y - Py$ and $\operatorname{span} C_c$. Let $\tilde{w}_t = Py - p_t = \tilde{w}_{t-1} - \alpha_t(x_t - Py + \tilde{w}_{t-1})$ and assume that $\tilde{w}_{t-1} \neq 0$. In short, if C consists of a single element then there is nothing to show and, otherwise, we have for $t \geq 2$ that $\alpha_t = \tilde{\alpha}_t$ and

$$\|\tilde{w}_t\|^2 = \|\tilde{w}_{t-1}\|^2 - 2\alpha_t \langle \tilde{w}_{t-1}, \tilde{w}_{t-1} + (x_t - Py)\rangle + \alpha_t^2 \|x_t - p_{t-1}\|^2 .$$

Observe that $\langle w_t, \tilde{w}_t\rangle = \langle y - p_t, Py - p_t\rangle = \|\tilde{w}_t\|^2$ because of the orthogonality and because $Py - p_t \in \operatorname{span} C_c$. This, together with the orthogonality between $y - Py$ and $x_t - Py$ yields

$$\langle w_{t-1}, w_{t-1} + (x_t - y)\rangle = \langle w_{t-1}, Py - p_{t-1} + x_t - Py\rangle = \langle \tilde{w}_{t-1}, \tilde{w}_{t-1} + x_t - Py\rangle .$$

Furthermore, $\|w_{t-1} + x_t - y\| = \|x_t - p_{t-1}\|$ and, hence,

$$\alpha_t = \frac{\langle \tilde{w}_{t-1}, \tilde{w}_{t-1} + (x_t - Py)\rangle}{\|x_t - p_{t-1}\|^2}.$$

By filling in this value of α_t we gain

$$\begin{aligned}\|\tilde{w}_t\|^2 &= \|\tilde{w}_{t-1}\|^2 - 2\frac{|\langle \tilde{w}_{t-1}, \tilde{w}_{t-1} + x_t - Py\rangle|^2}{\|x_t - p_{t-1}\|^2} + \frac{|\langle \tilde{w}_{t-1}, \tilde{w}_{t-1} + x_t - Py\rangle|^2}{\|x_t - p_{t-1}\|^2} \\ &= \frac{\|\tilde{w}_{t-1}\|^2 \|Py - x_t\|^2 - |\langle \tilde{w}_{t-1}, Py - x_t\rangle|^2}{\|x_t - p_{t-1}\|^2}.\end{aligned}$$

Since $\arg\max_{x\in S} \langle w_{t-1}, x\rangle = \arg\max_{x\in S} \langle \tilde{w}_{t-1}, x\rangle$ we have that

$$\langle \tilde{w}_{t-1}, x_t - Py\rangle = \max_{x\in C} \langle \tilde{w}_{t-1}, x - Py\rangle \geq \delta_F \|\tilde{w}_{t-1}\|$$

and

$$\|\tilde{w}_{t-1}\|^2 \|Py - x_t\|^2 - |\langle \tilde{w}_{t-1}, Py - x_t\rangle|^2 \leq \|\tilde{w}_{t-1}\|^2 (\|Py - x_t\|^2 - \delta_F^2).$$

Eq. 4 tells us now that $\|x_t - Py + (Py - p_{t-1})\|^2 \geq \|x_t - Py\|^2$ and

$$\|\tilde{w}_t\|^2 \leq \frac{\|\tilde{w}_{t-1}\|^2 (\|x_t - Py\|^2 - \delta_F^2)}{\|x_t - Py\|^2} \leq (1 - \delta_F^2/\mathrm{diam}^2(C)) \|\tilde{w}_{t-1}\|^2.$$

This is the second part of our claim. The first part follows directly from the particular form that α_t attains. With $\tilde{w}_{t-1} = 0$ we have

$$\alpha_t = \frac{\langle \tilde{w}_{t-1}, \tilde{w}_{t-1} + (x_t - Py)\rangle}{\|x_t - p_{t-1}\|^2} = 0$$

and $\tilde{w}_t = \tilde{w}_{t-1} = 0$. Finally, both cases imply the fast rate of convergence: Let $\gamma = \delta_F^2/\mathrm{diam}^2(C) \in (0, 1)$ then in both cases $\|\tilde{w}_t\|^2 \leq (1 - \gamma)\|\tilde{w}_{t-1}\|^2$ for all $t \geq 2$ and Lemma A.1(ii) from Beck and Teboulle (2004) tells us that $\|\tilde{w}_t\|^2 \leq e^{-\gamma t}\|\tilde{w}_1\|^2$. But then with $\beta = \gamma/2$ we have $\|Py - p_t\| \leq e^{-\beta t}\|\tilde{w}_1\|$ and the result follows. ❑

(ii) We address now $F \neq C$. First, we can observe that in the general case either $\tilde{\alpha}_t \leq 1$ holds after at most finite many steps or there is one final step where $\tilde{\alpha}_t > 1$ and the approximation error becomes zero afterwards, i.e. there exists a time $t_0 < \infty$ such that $\tilde{\alpha}_t \leq 1$ for all $t \geq t_0$ or, if $\tilde{\alpha}_t > 1$ for some $t \geq t_0$, then for all $s > t$ we have that $p_s = Py$ and $\tilde{\alpha}_s \in [0, 1]$. t_0 depends here on y and S. **P** We expand the argument of (i.a). In the case that S does not contain the element Py we can argue in the following way: If $\|y - Py\| > 0$ and $\|p_{t-1} - Py\| \leq (\|x_t - Py\|^2 - \langle p_{t-1} - Py, x_t - Py\rangle)/\|y - Py\|$ then

$$\begin{aligned}&\langle y - p_{t-1}, x_t - p_{t-1}\rangle = \|y - p_{t-1}\|^2 + \langle y - p_{t-1}, x_t - y\rangle \\ &= \|y - p_{t-1}\|^2 + \langle y - Py, x_t - y\rangle + \langle Py - p_{t-1}, Py - y\rangle + \langle Py - p_{t-1}, x_t - Py\rangle \\ &= \|Py - p_{t-1}\|^2 - \langle p_{t-1} - Py, x_t - Py\rangle + \langle y - Py, Py - p_{t-1}\rangle + \langle y - Py, x_t - Py\rangle \\ &\leq \|Py - p_{t-1}\|^2 - \langle p_{t-1} - Py, x_t - Py\rangle + \|y - Py\| \|p_{t-1} - Py\| \\ &\leq \|x_t - p_{t-1}\|^2\end{aligned}$$

and $\tilde{\alpha}_t \leq 1$. S consists of finite many elements and $\eta := \mathrm{dist}(S, Py) = \min_{x\in S} \|x - Py\| > 0$. Since $x_t \in S$ we know that $\|x_t - Py\| \geq \eta$. Furthermore,

$$\langle Py - p_{t-1}, x_t - Py\rangle = -\langle y - Py, x_t - Py\rangle + \langle y - p_{t-1}, x_t - Py\rangle \geq 0$$

because the first inner product is non-positive and the second term is non-negative (the same argument as in Theorem 2 (ii.e)). Hence,

$$\|x_t - Py\|^2 - \langle p_{t-1} - Py, x_t - Py\rangle \geq \eta^2 > 0.$$

Standard results for the CGM tell us that we have a constant $c > 0$ such that $\|p_t - y\|^2 - \|Py - y\|^2 \leq c/t$ and hence

$$\|p_t - Py\|^2 = \|p_t - y\|^2 + 2\langle p_t - y, y - Py\rangle + \|y - Py\|^2 \leq \|p_t - y\|^2 - \|y - Py\|^2 \leq c/t$$

where we used that

$$\langle p_t - y, y - Py\rangle = \langle p_t - Py, y - Py\rangle - \|y - Py\|^2 \leq -\|y - Py\|^2 .$$

Hence, for all $t \in \mathbb{N}$ with $t \geq t_0$, where $t_0 = c\,\|y - Py\|^2/\eta^4$, we know that $\tilde{\alpha}_t \leq 1$. If $y = Py$ then the argument simplifies to

$$\langle y - p_{t-1}, x_t - p_{t-1}\rangle \leq \|Py - p_{t-1}\|^2 - \langle p_{t-1} - Py, x_t - Py\rangle \leq \|x_t - p_{t-1}\|^2 .$$

The remaining case is the case where $Py \in S$. In fact, the only critical case is where $x_t = Py$. We can argue in the following way:

$$\tilde{\alpha}_t = \frac{\langle y - Py, Py - p_{t-1}\rangle + \langle Py - p_{t-1}, Py - p_{t-1}\rangle}{\|Py - p_{t-1}\|^2} \geq 1$$

and $\alpha_t = 1$. Hence, $p_t = Py$ and $x_{t+1} = \arg\max_{x\in S}\langle y - p_t, x\rangle = \arg\max_{x\in S}\langle y - Py, x - Py\rangle$. If $x_{t+1} = Py$ then $p_{t+1} = Py$ and $\tilde{\alpha}_{t+1} = 1$. Otherwise, x_{t+1} is an element of the minimal face. Hence, if $p_t = Py$ and $x_{t+1} \neq Py$ then

$$\tilde{\alpha}_{t+1} = \frac{\langle y - Py, x_{t+1} - p_t\rangle + \langle Py - p_t, x_{t+1} - p_t\rangle}{\|x_{t+1} - p_t\|^2} = 0 = \alpha_{t+1}$$

and $p_{t+1} = Py$. The same argument yields that $p_s = Py$ for all $s \geq t$ and $\tilde{\alpha}_s \in [0, 1]$. ❑

(iii) Consider now the split $\operatorname{span} C_c = U_F \oplus U_\delta \oplus U_b$. We will use here the same notation δ_m, δ_F etc. as in Theorem 2.

(a) If Assumption 1 holds then there exists a $s_0 < \infty$ such that for all $t \geq s_0$ the algorithm chooses elements $x_t \in F$. **P** Assumption 1 provides us with a time s_0 after which only elements in $D(\{x_t\}) \cup F$ are chosen. As in Theorem 2 (iv) we can observe that there exists a constant $c > 0$ such that for all $t \geq s_0$

$$\langle x_t - Py, (P_F + P_\delta)w_t\rangle \geq c\|(P_F + P_\delta)w_t\|.$$

$x_t \in S$ is here the element chosen at step t. Hence, the sequence $\{\|t(P_F + P_\delta)w_t\|\}_{t\geq 0}$ is bounded. Furthermore, because $0 \leq \langle x_t - Py, tw_t\rangle = \langle x_t - Py, t(P_F + P_\delta)w_t\rangle + \langle x_t - Py, tP_b w_t\rangle$ we can infer that

$$-\langle x_t - Py, tP_b w_t\rangle \leq \langle x_t - Py, t(P_F + P_\delta)w_t\rangle \leq \|x_t - Py\|\,\|t(P_F + P_\delta)w_t\|$$

and the sequence $\{-\langle x_t - Py, tP_b w_t\rangle\}_{t\geq 0}$ is bounded. But this implies that $x = x_t$ is either an element of F or it is selected only finitely often (and, hence, $x \notin D(\{x_t\})$). Therefore, $D(\{x_t\})$ is empty and the result follows. ❑

(b) There exists a constant $d > 0$ and a time u_0 after which for all $t \geq u_0$

$$\|P_F \tilde{w}_t\|^2 \leq \left(1 - \frac{\delta_F^2}{\operatorname{diam}^2(F)}\right) \|P_F \tilde{w}_{t-1}\|^2 + d \, \|(I - P_F)\tilde{w}_{t-1}\|^2 .$$

P Let us consider $t > t_0 \vee s_0$ with s_0 from (a) and t_0 from (ii). We know that only elements in F are chosen at t and hence

$$\begin{aligned}\|x_t - p_{t-1}\|^2 \tilde{\alpha}_t &= \langle w_{t-1}, \tilde{w}_{t-1} + x_t - Py \rangle = \langle \tilde{w}_{t-1}, \tilde{w}_{t-1} + x_t - Py \rangle + \langle y - Py, \tilde{w}_{t-1} \rangle \\ &= \langle P_F \tilde{w}_{t-1}, P_F \tilde{w}_{t-1} + x_t - Py \rangle + \|(I - P_F)\tilde{w}_{t-1}\|^2 + \langle y - Py, \tilde{w}_{t-1} \rangle .\end{aligned}$$

Also, $\tilde{\alpha}_t = \alpha_t$ and

$$\begin{aligned}\|P_F \tilde{w}_t\|^2 &= \|P_F \tilde{w}_{t-1}\|^2 - 2\alpha_t \, \langle P_F \tilde{w}_{t-1}, P_F \tilde{w}_{t-1} + x_t - Py \rangle + \alpha_t^2 \, \|x_t - P_F p_{t-1}\|^2 \\ &\leq \left(1 - \frac{\delta_F^2}{\operatorname{diam}^2(F)}\right) \|P_F \tilde{w}_{t-1}\|^2 + \frac{(\|(I - P_F)\tilde{w}_{t-1}\|^2 + \langle y - Py, \tilde{w}_{t-1} \rangle)^2}{\|x_t - p_{t-1}\|^2} .\end{aligned}$$

Now, since p_t converges to Py there exists a time $u_0 > t_0 \vee s_0$ after which $\|x_t - p_{t-1}\| \geq \|x_t - Py\|/2 \geq \min_{x \in S} \|x - Py\|/2$ and $\|(I - P_F)\tilde{w}_{t-1}\|^2 \leq \|(I - P_F)\tilde{w}_{t-1}\| \leq 1$. Hence, for all $t \geq u_0$ we have that

$$\|P_F \tilde{w}_t\|^2 \leq \left(1 - \frac{\delta_F^2}{\operatorname{diam}^2(F)}\right) \|P_F \tilde{w}_{t-1}\|^2 + \frac{4(1 + \|y - Py\|)}{\min_{x \in S} \|x - Py\|^2} \, \|(I - P_F)\tilde{w}_{t-1}\|^2 .$$

Choosing $d = 4(1 + \|y - Py\|)/(\min_{x \in S} \|x - Py\|^2)$ yields the result. **Q**

(c) If $\min_{x \in S} \|x - Py\| > 0$ and if $\|(I - P_F)\tilde{w}_{t_0 \vee s_0}\| > 0$ with s_0 from (a) and t_0 from (ii) then the sequence $\{\|(I - P_F)\tilde{w}_t\|\}_{t \geq 1}$ converges sub-linearly. **P** Let us consider $t \geq t_0 \vee s_0$. We know that only elements in F are chosen at t and that $\alpha_t = \tilde{\alpha}_t$. Therefore $\|(I - P_F)\tilde{w}_{t+1}\| = (1 - \alpha_{t+1}) \, \|(I - P_F)\tilde{w}_t\|$ and

$$\begin{aligned}\frac{\|(I - P_F)\tilde{w}_{t+1}\|}{\|(I - P_F)\tilde{w}_t\|} = 1 - \alpha_{t+1} &= \frac{\|x_{t+1} - Py + \tilde{w}_t\|^2 - \langle w_t, \tilde{w}_t + x_{t+1} - Py \rangle}{\|x_{t+1} - p_t\|^2} \\ &= \frac{\|x_{t+1} - Py\|^2 + \langle P_F \tilde{w}_t, x_{t+1} - Py \rangle - \langle y - Py, P_b \tilde{w}_t \rangle}{\|x_{t+1} - Py\|^2 + 2 \langle x_{t+1} - Py, \tilde{w}_t \rangle + \|\tilde{w}_t\|^2}\end{aligned}$$

which converges to 1 since $\lim_{t \to \infty} \|\tilde{w}_t\| = 0$. This means that the sequence converges sub-linearly. **Q** ■

6.3 Corollaries

Corollary 4 *Given a compact convex set $C \subset \mathcal{H}$ and a finite subset S of C with $\operatorname{ex} C \subseteq S$. If for $y \in \mathcal{H}$ there exists a decomposition $U_F \oplus U_\delta \oplus U_b$ of $\operatorname{span} C_c$ such that U_b is one dimensional then Assumption 1 is fulfilled. In particular, in this case there exists a constant $b > 0$ such that Algorithm 1 has a worst-case approximation error of b/t for all $t \geq 1$.*

Proof Consider elements $x, x' \in S \backslash F$. Since $\|P_b(x - Py)\|, \|P_b(x' - Py)\| > 0$ we know that $\Delta = \|P_b(x - Py)\| / \|P_b(x' - Py)\| < \infty$ and since U_b is one-dimensional we have for any t

$$\langle -P_b w_t, x - Py \rangle = \|P_b w_t\| \, \|P_b(x - Py)\| \leq \Delta \, \|P_b w_t\| \, \|P_b(x' - Py)\| = \Delta \langle -P_b w_t, x' - Py \rangle$$

and Assumption 1 is fulfilled. ∎

Let for the following corollary $A = \{z \in \mathcal{H} : z = \alpha P_C(y - Py) + P_C Py, \alpha \in [0, \infty)\}$.

Corollary 5 *Given a compact convex set $C \subset \mathcal{H}$ and a finite subset S of C with* $\operatorname{ex} C \subseteq S$. *Assumption 1 is fulfilled if $P_C y \in \mathcal{H} \backslash C$ and whenever $Pz = Py$ for some $z \in \mathcal{H}$ then $P_C z \in A$ holds. In particular, in this case there exists a constant $b > 0$ such that Algorithm 1 has a worst-case approximation error of b/t for all $t \geq 1$.*

Proof The assumption guarantees us that U_b is one-dimensional. Assume otherwise. By assumption $\|P_C(y - Py)\| > 0$ and we can chose E_b such that there exist two elements e_1, e_2 in E_b with $e_1 = P_C(y - Py)/\|P_C(y - Py)\|$ and $\langle e_1, e_2 \rangle = 0, \|e_1\| = \|e_2\| = 1$. We claim that $z_1 = Py + e_1$ and $z_2 = Py + e_2$ are both projected onto Py, i.e. $Pz_1 = Py = Pz_2$: We know that $P_{e_1}[C_c] \subset (-\infty, 0]$ and $\langle e_1, x - Py \rangle \leq 0$ for all $x \in C$. Hence, for all $x \in C$ we know that $\langle z_1 - Py, x - Py \rangle = \langle e_1, x - Py \rangle \leq 0$. But, this means that $Pz_1 = Py$. The same argument applies to z_2 and Py is the projection of two elements for which $\langle z_1 - Py, z_2 - Py \rangle = 0$. Furthermore, $P_C z_1 = P_C Py + e_1$ and $P_C z_2 = P_C Py + e_2$. If $P_C z_2 \in A$ then there exist $\alpha, \tilde{\alpha} > 0$,

$$e_2 + P_C Py = P_C z_2 = \alpha P_C(y - Py) + P_C Py = \tilde{\alpha} e_1 + P_C Py$$

and $e_2 = \tilde{\alpha} e_1$ with a contradiction to orthogonality. ∎

For the next corollaries let $d = \dim U_b$, let $e_1, \ldots, e_d$ be any basis of U_b and let $r = \sup_{x \in C} \|x\|$. Also introduce $\alpha = \min_{i \leq d} \min\{|\langle e_i, x - Py \rangle| : x \in S \backslash F, \langle e_i, x - Py \rangle \neq 0\} > 0$.

Corollary 6 *Let C be a compact convex set in some Hilbert space $\mathcal{H}$, S a finite set with* $\operatorname{cch} S = C$, *and $y \in \mathcal{H}$ such that there exists a split into $U_F \oplus U_\delta \oplus U_b$ of* $\operatorname{span} C_c$ *with $U_\delta = \{0\}$. Assumption 1 is fulfilled and there exists a constant $b > 0$ such that Algorithm 1 has a worst-case approximation error of b/t for all $t \geq 1$. The constant b can be chosen as $\sqrt{d} 4r^3/(\alpha \delta_F) + 6r^2(1/\delta_F + 1/\alpha) + 5r$.*

Proof If $F_c = \{0\}$ then each element $x \in S \backslash F$ can only be chosen once since $0 > \langle w_t, x - Py \rangle = \langle P_b w_t, x - Py \rangle$ if x has been chosen at least once. This implies that $D(\{x_t\})$ is, in fact, the empty set and Assumption 1 is fulfilled. It is also easy to see that $\langle w_t, e_i \rangle \leq \|x - Py\| \leq 2r$ for all $i \leq d$. This implies that the constant b can in this case be chosen as $2r\sqrt{d}$. Also, since δ_F and α are upper bounded by r this implies that $b \leq 2r^3\sqrt{d}/(\alpha \delta_F)$.

Assume now that $F_c \neq \{0\}$ and that there exists a $x \in D(\{x_t\})$. By definition $x \in S \backslash F$ and x is chosen infinitely often. Hence, $\{\langle P_b w_t, x - Py \rangle\}_{t \geq 1}$ is a non-increasing sequence that diverges to $-\infty$. There exists a $\delta_F > 0$ such that for any w_t, $\max_{x' \in S \cap F} \langle w_t, x' - Py \rangle \geq \delta_F \|P_F w_t\|$. Also, for any $x' \in S$, we have that $\langle w_t, x' - Py \rangle = \langle P_F w_t, x' - Py \rangle + \langle P_b w_t, x' - Py \rangle$ and, since the term $\langle P_b w_t, x' - Py \rangle$ is always non-positive, we know that if $P_F w_t \neq 0$ elements will be chosen such that $\langle P_F w_t, x' - Py \rangle \geq \delta_F \|P_F w_t\|$. Hence, whenever $\|P_F w_t\| \geq 2r^2/\delta_F$ for some t then $\|P_F w_{t+1}\| \leq \|P_F w_t\|$. This implies that $\|P_F w_t\| \leq 2r^2/\delta_F + 3r$ for all $t \geq 1$. Hence, for x to be chosen infinitely often we need for all time steps t where x is chosen that

$$0 \leq \langle w_t, x - Py \rangle \leq \langle P_F w_t, x - Py \rangle + \langle P_b w_t, x - Py \rangle \leq 2r(2r^2/\delta_F + 3r) + \langle P_b w_t, x - Py \rangle$$

since $\|x - Py\| \leq 2r$. Therefore, $-\langle P_b w_t, x - Py\rangle \leq r(4r^2/\delta_F + 6r)$ and $\{\langle P_b w_t, x - Py\rangle\}_{t\geq 1}$ cannot diverge to $-\infty$. In other words, x cannot be chosen infinitely often with a contradiction to the initial assumption. We can also control the constant in this case: $\|P_b w_t\|$ can only grow if either $P_F w_t = 0$ or there is an $x' \in S\backslash F$ such that $-\langle P_b w_t, x' - Py\rangle \leq r(4r^2/\delta_F + 6r)$. Then for w_t if for any $i \leq d$

$$\alpha \langle w_t, e_i\rangle > r(4r^2/\delta_F + 6r)$$

then no $x \in S\backslash F$ with $\langle e_i, x - Py\rangle \neq 0$ can be played since for such a x

$$-\langle P_b w_t, x - Py\rangle \geq \langle e_i, x - Py\rangle \langle e_i, w_t\rangle > r(4r^2/\delta_F + 6r).$$

Hence, $\langle e_i, w_t\rangle \leq r(4r^2/\delta_F + 6r)/\alpha + 2r$ since the increment of w_t in one step is bounded by $2r$. Since this holds for each $i \leq d$ we gain the bound $\|P_b w_t\| \leq \sqrt{d}4r^3/(\delta_F\alpha) + 6r^2/\alpha + 2r$ and

$$\|w_t\| \leq \|P_F w_t\| + \|P_b w_t\| \leq \sqrt{d}4r^3/(\delta_F\alpha) + 6r^2(1/\delta_F + 1/\alpha) + 5r.$$

■

Corollary 7 *Let $y \in \mathbb{R}^d$, Q any orthogonal matrix, $c > 0$ any scaling, $z \in \mathbb{R}^d$, $S = \{0,1\}^d$ and $C = [0,1]^d$, $d \geq 1$. Assumption 1 is fulfilled for the set $\tilde{S} = cQ[S + z]$ and $\tilde{C} = cQ[C + z]$ (independently of the dimensionality of the face Py lies in). In particular, there exists a constant $b > 0$ such that Algorithm 1 has a worst-case approximation error of b/t for all $t \geq 1$. The constant b can be chosen as $\sqrt{d}4r^3/(c\delta_F) + 6r^2(1/\delta_F + 1/c) + 5r$.*

Proof Because the algorithm does not change by translating and rotating S and C we can consider a hypercube anchored at the origin and aligned with the standard basis of $\mathbb{R}^d$. We therefore assume without loss of generality that we have to deal with a scaled standard hypercube in $\mathbb{R}^d$. Let us assume in the following that $\dim U_F = k \leq d$.

Observe that either $U_F = \{0\}$ or we can write $U_F = \operatorname{span}\{e_{i_1}, \ldots, e_{i_k}\}$, where $e_1, \ldots, e_d$ is the standard basis in $\mathbb{R}^d$, $i_j \leq d$ for all $j \leq k$, and $k = \dim U_F$. This holds because a k-dimensional face of the hypercube is a translated k-dimensional hypercube in a subspace spanned by some basis vectors $e_{i_1}, \ldots, e_{i_k}$. Define the following index sets $\mathcal{I} = \{i_1, \ldots, i_k\}$, $\mathcal{I} = \emptyset$ if $U_F = \{0\}$, and $\mathcal{J} = \{1, \ldots, d\}\backslash\mathcal{I}$.

There are signs $s_j \in \{-1, 1\}$, $j \in \mathcal{J}$, such that $E_b = \{s_j e_j : j \in \mathcal{J}\}$ is a basis for U_b, and the split into $U_F \oplus U_\delta \oplus U_b$ is satisfying the conditions of Assumption 1. In particular, $U_\delta = \{0\}$, $P_C(y - Py) \in U_b$ and for all $e \in E_b$ we have $\{\langle x, e\rangle : x \in C_c\} \subset (-\infty, 0]$. **P** $U_\delta = \{0\}$ because the minimal faces which are orthogonally projected onto a face that contains a ball around Py (relative to the projection) are translated versions of the minimal face and they induce the same subspace as the minimal face. That also implies directly that E_b can be chosen to fulfill Assumption 1. $P_C(y - Py)$ stands orthogonal on U_F and, hence, lies fully in U_b. Therefore, Corollary 6 applies. Observe that α can here be chosen as the scaling factor c and the corollary provides the stated result since $\dim U_b \leq d$. ■

Corollary 8 *Let $y \in \mathbb{R}^d$, $S = \{e_i : i \leq d\}$ and $C = \Delta^{d-1} = \operatorname{cch} S$, $d \geq 1$. Assumption 1 is fulfilled and there exists a constant $b > 0$ such that Algorithm 1 has a worst-case approximation error of b/t for all $t \geq 1$. The constant b can be chosen as $4dr^3/\delta_F + 6r^2(1/\delta_F + \sqrt{d}) + 5r$.*

Proof Due to the rotation invariance it suffices to consider the case where $F = \operatorname{cch}\{e_1, \ldots, e_k\}$ for some $k \leq d$. Let us consider first the case where $F = \{e_1\}$ and, hence, $F_c = \{0\}$. In this case, $C_c = \operatorname{cch}\{0, e_2 - e_1, \ldots, e_d - e_1\}$. Observe that $\langle x + e_1, e_j \rangle \geq 0$ for any $x \in C_c$ and all $j \leq d$ since $x + e_1 \in C$ and, hence, $x + e_1 = \sum_{i \leq d} \alpha_i e_i$ for some non-negative α_i. In particular, for $j \geq 2$, $\langle x, e_j \rangle \geq 0$ for all $x \in C_c$. Furthermore, $\langle x, e_1 \rangle \leq 0$ for all $x \in C_c$ because $x + e_1 = \sum_{i \leq d} \alpha_i e_i$ with non-negative scalars that fulfill $\sum_{i \leq d} \alpha_i = 1$. I.e. $\langle x, e_1 \rangle = \langle x + e_1, e_1 \rangle - 1 \leq \|e_1\|^2 - 1 = 0$. Therefore, we have an orthonormal basis $\{-e_1, e_2, \ldots, e_d\}$, of $\mathbb{R}^d$ such that $\langle x, e \rangle \geq 0$ for all basis elements e and all $x \in C_c$. This implies that there exists no basis element $\tilde{e} \in \mathbb{R}^d$ such that there exists elements $x, x' \in C_c$ with $\langle x, \tilde{e} \rangle > 0 > \langle x', \tilde{e} \rangle$ and, hence, $\{-e_1, e_2, \ldots, e_d\}$ spans the d-dimensional space U_b and $U_\delta = \{0\}$. The rate of convergence follows now from Corollary 6.

Let us consider the case $F = \operatorname{cch}\{e_1, \ldots, e_k\}$ and $k \geq 2$. For any e_j, $j > k$, and any $x \in C_c$, $\langle x + Py, e_j \rangle \geq 0$ since $x + Py = \sum_{i \leq d} \alpha_i e_i$ for some non-negative α_i. But this implies $\langle x, e_j \rangle = \langle x + Py, e_j \rangle \geq 0$ because Py is a convex combination of $e_1, \ldots, e_k$. Also $e_{k+1}, \ldots, e_d \in F_c^\perp$ because $\operatorname{span} F_c = \operatorname{span}\{e_2 - e_1, \ldots, e_k - e_1\}$ and $\langle e_i - e_1, e_j \rangle = 0$ for any $i \leq k, j > k$. Finally, consider $e = \sum_{j=1}^k e_j / \sqrt{k}$, $\|e\| = 1$, and observe that $\langle e, e_i \rangle = 0$ for all $i > k$. Also, $\langle e, e_i - e_1 \rangle = 0$ for $i \leq k$. This implies that e stands orthogonal on F_c because for $v \in \operatorname{span} F_c$, $v = \sum_{i=2}^k \alpha_i (e_i - e_1), \alpha_i \in \mathbb{R}$, we have that $\langle e, v \rangle = \sum_{i=2}^k \alpha_i \langle e_i - e_1, e_i + e_1 \rangle = 0$. Furthermore, for any $x \in C_c$, $x = \sum_{i=1}^d \alpha_i e_i - Py$, $\sum_{i=1}^d \alpha_i = 1$,

$$\langle e, x \rangle = \sum_{i=1}^d \alpha_i \langle e, e_i - Py \rangle = \sum_{i=1}^d \alpha_i \langle e, e_i - e_1 \rangle - \langle e, Py - e_1 \rangle = 0.$$

Hence, we have $d + 1 - k$ orthonormal vectors $e_{k+1}, \ldots, e_d, e$ that lie in the orthogonal complement of F_c and for any $x \in C_c$, $\langle x, e_i \rangle \geq 0, \langle x, e \rangle \geq 0$, for all $k + 1 \leq i \leq d$. Hence, $U_\delta = \{0\}$ and the result on the rate of convergence follows from Corollary 6.

The constant also follows directly from the corollary: in case that $F_c = \{0\}$ we have that $S \setminus F = \{e_2, \ldots, e_d\}$, the basis of U_b is $\{-e_1, e_2, \ldots, e_d\}$ and $Py = e_1$. α can be lower bounded by $1 : \langle -e_1, x - Py \rangle = 1$ for any $x \in S \setminus F$; $\langle e_i, e_j - Py \rangle$ attains value 1 if $i = j \geq 2$ and value 0 if $i \neq j, i, j \geq 2$.

Otherwise, $S \setminus F = \{e_{k+1}, \ldots, e_d\}$, the basis of U_b is $\{e_{k+1}, \ldots, e_d, \sum_{i=1}^k e_i / \sqrt{k}\}$ and $Py = \sum_{i=1}^k \alpha_i e_i$ for some $\alpha_i \geq 0, \sum_{i=1}^k \alpha_i = 1$. Hence, $\langle e_i, e_j - Py \rangle = 0$ if $i \neq j, i, j > k$; $\langle e_i, e_i - Py \rangle = 1$ if $i > k$; $\langle \sum_{i=1}^k e_i / \sqrt{k}, e_j - Py \rangle = 1/\sqrt{k}$, where $j \geq k$. Hence, $\alpha \geq 1/\sqrt{d}$ and the result follows. ■

References

N. Aronszajn. Theory of reproducing kernels. *Trans. of the American Mathematical Society*, 1950.

F. Bach, S. Lacoste-Julien, and G. Obozinski. On the equivalence between herding and conditional gradient algorithms. In *International Conference on Machine Learning*, 2012.

A. Beck and M. Teboulle. A conditional gradient method with linear rate of convergence for solving convex linear systems. *Math. Meth. Op. Res.*, 2004.

M.D. Cannon and C.D. Cullum. A tight upper bound on the rate of convergence of the frank-wolfe algorithm. *SIAM J. Control Optim.*, 1968.

Y. Chen, M. Welling, and A. Smola. Supersamples from kernel-herding. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2010.

K. L. Clarkson. Coresets, sparse greedy approximation,and the frank-wolfe algorithm. *ACM Transactions on Algorithms*, 2010.

J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *International Conference on Machine Learning*, 2008.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Res. Logist. Quart.*, 1956.

D. Garber and E. Hazan. A polynomial time conditional gradient algorithm with applications to online and stochastic optimization. *CoRR*, abs/1301.4666, 2013.

D. Garber and E. Hazan. Faster rates for the frank-wolfe method over strongly-convex sets. In *International Conference on Machine Learning*, 2015.

J. Guélat and P. Marcotte. Some comments on wolfe's 'away step'. *Mathematical Programming*, 1986.

M. Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International Conference on Machine Learning*, 2013.

S. Lacoste-Julien and M. Jaggi. On the global linear convergence of frank-wolfe optimization variants. In *Advances in Neural Information Processing*, 2015.

R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B.*, 1996.

K.C. Toh, M.J. Todd, and R.H. Tutuncu. Sdpt3 — a matlab software package for semidefinite programming. *Optimization Methods and Software*, 1999.

I. Tsang, J. Kwok, and P.-M. Cheung. Core vector machines: Fast svm training on very large data sets. *The Journal of Machine Learning Research*, 2005a.

I. Tsang, J. Kwok, and K. Lai. Core vector regression for very large regression problems. In *International Conference on Machine Learning*, 2005b.

R.H Tutuncu, K.C. Toh, and M.J. Todd. Solving semidefinite-quadratic-linear programs using sdpt3. *Mathematical Programming Ser. B*, 2003.

M. Welling. Herding dynamical weights to learn. In *International Conference on Machine Learning*, 2009.

P. Wolfe. *Convergence theory in nonlinear programming*, chapter 1. North-Holland Publishing Company, 1970.

P. Wolfe. Finding the nearest point in a polytope. *Mathematical Programming*, 1976.

Y. Zhang, J. Duchi, and M. Wainwright. Divide and conquer kernel ridge regression. In *Conference on Learning Theory*, 2013.