Adaptive Processing in a World of Projections

Sergios Theodoridis[1]

Joint work with
Konstantinos Slavakis[2] and Isao Yamada[3]

[1]University of Athens, Greece

[2]University of Peloponnese, Greece

[3]Tokyo Institute of Technology, Japan

January 16, 2009

"ΟΥΔΕΙΣ ΑΓΕΩΜΕΤΡΗΤΟΣ ΕΙΣΙ"

“ΟΥΔΕΙΣ ΑΓΕΩΜΕΤΡΗΤΟΣ ΕΙΣΙ”

(“Those who do not know geometry are not welcome here”)

Plato’s Academy of Philosophy

Outline

- The fundamental tool of metric projections in Hilbert spaces.
- The Set Theoretic Estimation approach and multiple intersecting closed convex sets.
- Online classification and regression in Reproducing Kernel Hilbert Spaces (RKHS).
- Incorporating a-priori constraints in the design.
- An algorithmic solution to constrained online learning in RKHS.
- A nonlinear adaptive beamforming application.

Problem Definition

Given

- A set of measurements $(\boldsymbol{x}_n, y_n)_{n=1}^{N}$, which are jointly distributed, and
- A parametric set of functions

$$\mathcal{F} = \{f_\alpha(\boldsymbol{x}) : \alpha \in A \subset \mathbb{R}^k\}.$$

Compute an $f(\cdot)$ that best approximates y, given the value of $\boldsymbol{x}$:

$$y \approx f(\boldsymbol{x}).$$

Problem Definition

Given

- A set of measurements $(\boldsymbol{x}_n, y_n)_{n=1}^N$, which are jointly distributed, and
- A parametric set of functions

$$\mathcal{F} = \{f_\alpha(\boldsymbol{x}) : \alpha \in A \subset \mathbb{R}^k\}.$$

Compute an $f(\cdot)$ that best approximates y, given the value of $\boldsymbol{x}$:

$$y \approx f(\boldsymbol{x}).$$

Special Cases

Smoothing, prediction, filtering, system identification, beamforming, curve-fitting, regression, and classification.

The More Classical Approach

Select a loss function $\ell(\cdot,\cdot)$ and estimate $f(\cdot)$ so that

$$f(\cdot) \in \{f_\alpha(\cdot) \in \arg\min_\alpha \sum_{n=1}^{N} \ell(y_n, f_\alpha(\boldsymbol{x}_n))\}.$$

The More Classical Approach

Select a loss function $\ell(\cdot,\cdot)$ and estimate $f(\cdot)$ so that

$$f(\cdot) \in \{f_\alpha(\cdot) \in \arg\min_\alpha \sum_{n=1}^{N} \ell(y_n, f_\alpha(\boldsymbol{x}_n))\}.$$

Drawbacks

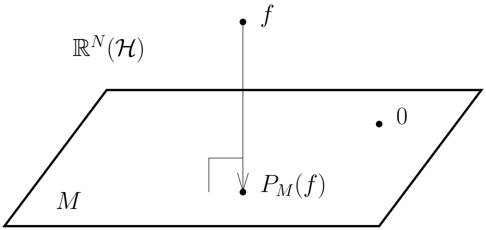
- Most often, in practice, the choice of the cost is dictated not by physical reasoning but by the computational tractability.
- The existence of a-priori information in the form of constraints makes the task even more difficult.
- The optimization task is solved iteratively, and iterations freeze after a finite number of steps. Thus, the obtained solution lies in a neighborhood of the optimal one.
- The stochastic nature of the data and the existence of noise add another uncertainty on the optimality of the obtained solution.

- In this talk we are concerned in finding a set of solutions that are in agreement with all the available information.

- This will be achieved in the general context of fixed point theory, using convex analysis and the powerful tool of projections.

Theorem

Given a Euclidean $\mathbb{R}^N$ or a Hilbert space $\mathcal{H}$, the projection of a point f onto a closed subspace M is the point $P_M(f) \in M$ that lies closest to f (Pythagoras Theorem).



Projection onto a Closed Convex Set

Theorem

Let C be a closed convex set in a Hilbert space $\mathcal{H}$. Then, for each $f \in \mathcal{H}$ there exists a unique $f_* \in C$ such that

$$\|f - f_*\| = \min_{g \in C} \|f - g\|.$$

Projection onto a Closed Convex Set

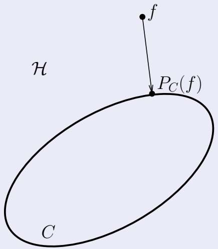
Theorem

Let C be a closed convex set in a Hilbert space $\mathcal{H}$. Then, for each $f \in \mathcal{H}$ there exists a unique $f_* \in C$ such that

$$\|f - f_*\| = \min_{g \in C} \|f - g\|.$$

Definition (Metric Projection Mapping)

Projection is the mapping $P_C : \mathcal{H} \to C : f \mapsto f_*$.



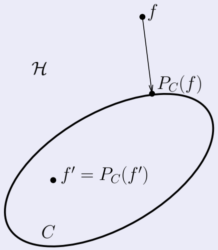
Theorem

Let C be a closed convex set in a Hilbert space $\mathcal{H}$. Then, for each $f \in \mathcal{H}$ there exists a unique $f_* \in C$ such that

$$\|f - f_*\| = \min_{g \in C} \|f - g\|.$$

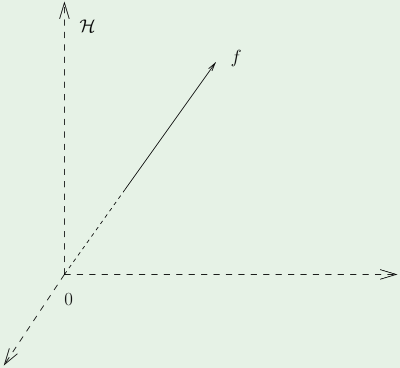
Definition (Metric Projection Mapping)

Projection is the mapping $P_C : \mathcal{H} \to C : f \mapsto f_*$.



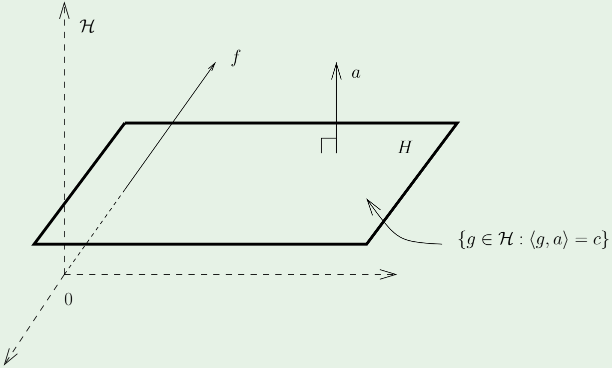
Projectors

Example (Hyperplane $H := \{g \in \mathcal{H} : \langle g, a \rangle = c\}$)

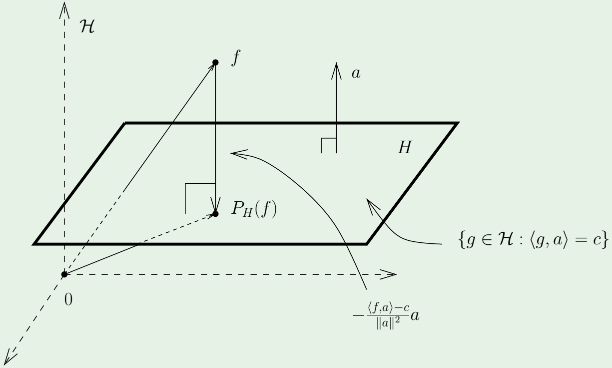


Projectors

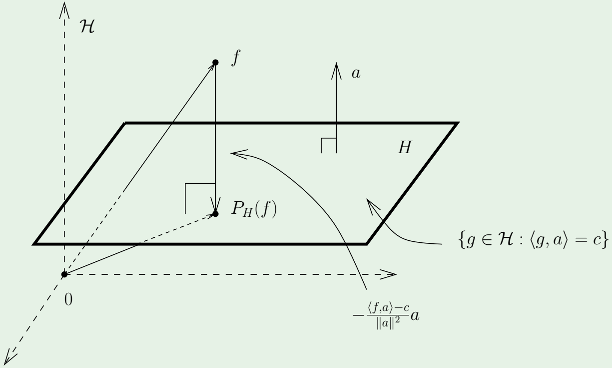
Example (Hyperplane $H := \{g \in \mathcal{H} : \langle g, a \rangle = c\}$)



Example (Hyperplane $H := \{g \in \mathcal{H} : \langle g, a \rangle = c\}$)

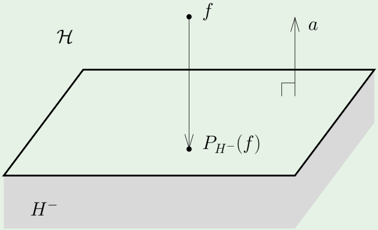


Example (Hyperplane $H := \{g \in \mathcal{H} : \langle g, a \rangle = c\}$)

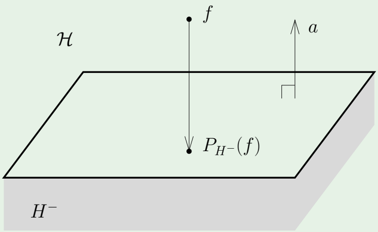


$$P_H(f) = f - \frac{\langle f, a \rangle - c}{\|a\|^2} a, \qquad \forall f \in \mathcal{H}.$$

Example (Halfspace $H^- := \{g \in \mathcal{H} : \langle g, a \rangle \leq c\}$)

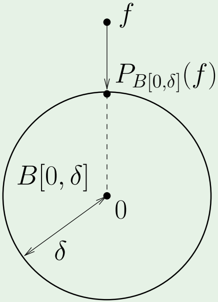


Example (Halfspace $H^- := \{g \in \mathcal{H} : \langle g, a \rangle \leq c\}$)

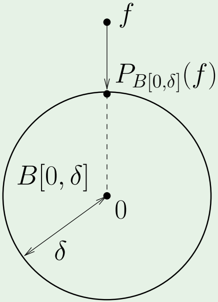


$$P_{H^-}(f) = f - \frac{\max\{0, \langle f, a \rangle - c\}}{\|a\|^2} a, \qquad \forall f \in \mathcal{H}.$$

Example (Closed Ball $B[0,\delta] := \{g \in \mathcal{H} : \|g\| \leq \delta\}$)

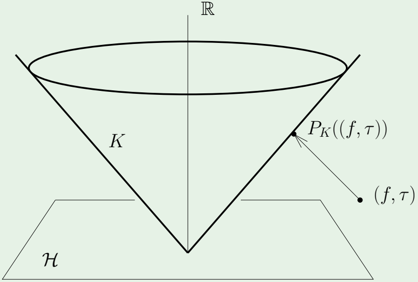


Example (Closed Ball $B[0,\delta] := \{g \in \mathcal{H} : \|g\| \leq \delta\}$)



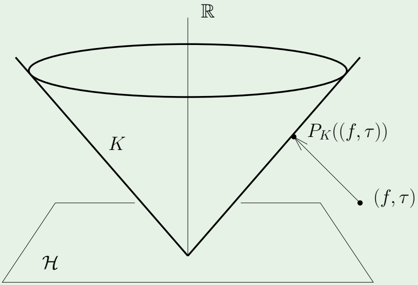
$$P_{B[0,\delta]}(f) := \begin{cases} f, & \text{if } \|f\| \leq \delta, \\ \frac{\delta}{\|f\|} f, & \text{if } \|f\| > \delta. \end{cases}, \qquad \forall f \in \mathcal{H}.$$

Example (Icecream Cone $K := \{(f,\tau) \in \mathcal{H} \times \mathbb{R} : \|f\| \geq \tau\}$)



Projectors

Example (Icecream Cone $K := \{(f,\tau) \in \mathcal{H} \times \mathbb{R} : \|f\| \geq \tau\}$)



$$P_K((f,\tau)) = \begin{cases} (f,\tau), & \text{if } \|f\| \leq \tau, \\ (0,0), & \text{if } \|f\| \leq -\tau, \\ \frac{\|f\|+\tau}{2}\left(\frac{f}{\|f\|}, 1\right), & \text{otherwise}, \end{cases} \qquad \forall (f,\tau) \in \mathcal{H} \times \mathbb{R}.$$

Definition

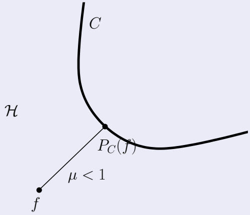
Given a closed convex set C and its associated projection mapping P_C, the relaxed projection mapping is defined as

$$T_C(f) := f + \mu(P_C(f) - f), \mu \in (0, 2), \quad \forall f \in \mathcal{H}.$$

Definition

Given a closed convex set C and its associated projection mapping P_C, the relaxed projection mapping is defined as

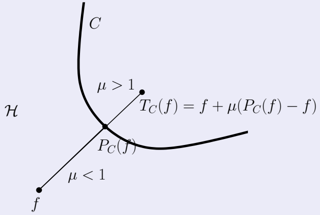
$$T_C(f) := f + \mu(P_C(f) - f), \mu \in (0, 2), \quad \forall f \in \mathcal{H}.$$



Definition

Given a closed convex set C and its associated projection mapping P_C, the relaxed projection mapping is defined as

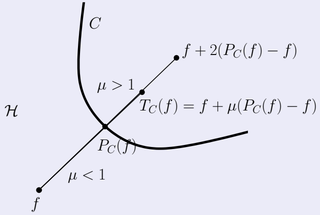
$$T_C(f) := f + \mu(P_C(f) - f), \mu \in (0, 2), \quad \forall f \in \mathcal{H}.$$



Definition

Given a closed convex set C and its associated projection mapping P_C, the relaxed projection mapping is defined as

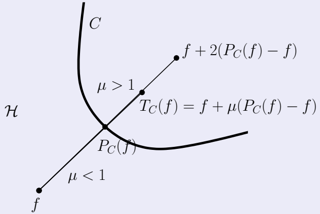
$$T_C(f) := f + \mu(P_C(f) - f), \mu \in (0, 2), \quad \forall f \in \mathcal{H}.$$



Definition

Given a closed convex set C and its associated projection mapping P_C, the relaxed projection mapping is defined as

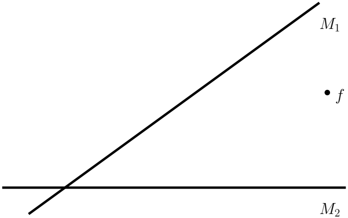
$$T_C(f) := f + \mu(P_C(f) - f), \mu \in (0, 2), \quad \forall f \in \mathcal{H}.$$



Remark: The use of the relaxed projection operator with $\mu > 1$ can, potentially, speed up the convergence rate of the algorithms to be presented.

Alternating Projections

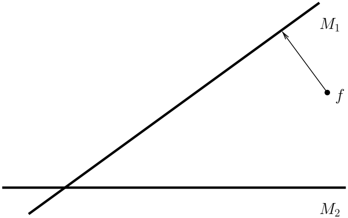
Composition of Projection Mappings: Let M_1 and M_2 be closed subspaces in the Hilbert space $\mathcal{H}$. For any $f \in \mathcal{H}$, define the sequence of projections:



Alternating Projections

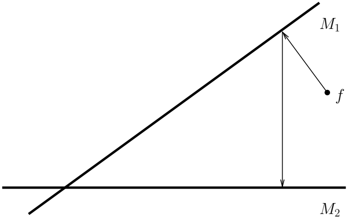
Composition of Projection Mappings: Let M_1 and M_2 be closed subspaces in the Hilbert space $\mathcal{H}$. For any $f \in \mathcal{H}$, define the sequence of projections:

$$P_{M_1}(f).$$



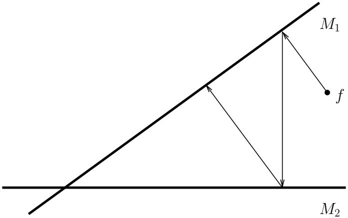
Composition of Projection Mappings: Let M_1 and M_2 be closed subspaces in the Hilbert space $\mathcal{H}$. For any $f \in \mathcal{H}$, define the sequence of projections:

$$P_{M_2} P_{M_1}(f).$$



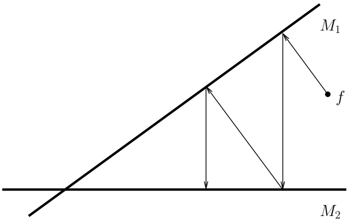
Composition of Projection Mappings: Let M_1 and M_2 be closed subspaces in the Hilbert space $\mathcal{H}$. For any $f \in \mathcal{H}$, define the sequence of projections:

$$P_{M_1} P_{M_2} P_{M_1}(f).$$



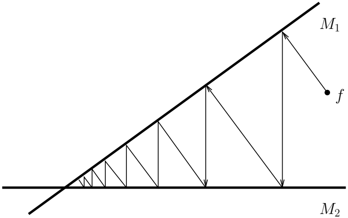
Composition of Projection Mappings: Let M_1 and M_2 be closed subspaces in the Hilbert space $\mathcal{H}$. For any $f \in \mathcal{H}$, define the sequence of projections:

$$P_{M_2} P_{M_1} P_{M_2} P_{M_1}(f).$$



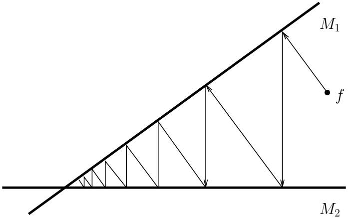
Composition of Projection Mappings: Let M_1 and M_2 be closed subspaces in the Hilbert space $\mathcal{H}$. For any $f \in \mathcal{H}$, define the sequence of projections:

$$\cdots P_{M_2} P_{M_1} P_{M_2} P_{M_1}(f).$$



Composition of Projection Mappings: Let M_1 and M_2 be closed subspaces in the Hilbert space $\mathcal{H}$. For any $f \in \mathcal{H}$, define the sequence of projections:

$$\cdots P_{M_2} P_{M_1} P_{M_2} P_{M_1}(f).$$



Theorem (Von Neumann '33)

For any $f \in \mathcal{H}$, $\quad \lim_{n\to\infty}(P_{M_2}P_{M_1})^n(f) = P_{M_1\cap M_2}(f).$

Projections Onto Convex Sets (POCS)

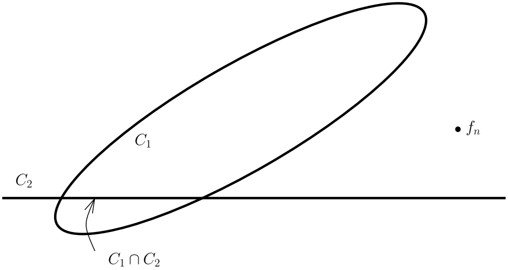
Given a finite number of closed convex sets $C_1, \ldots, C_q$, with $\bigcap_{i=1}^{q} C_i \neq \emptyset$, let their associated relaxed projection mappings be $T_{C_1}, \ldots, T_{C_q}$. For any $f_0 \in \mathcal{H}$, this defines the sequence of points

$$f_{n+1} := T_{C_q} \cdots T_{C_1}(f_n), \qquad \forall n.$$

Projections Onto Convex Sets (POCS)

Given a finite number of closed convex sets $C_1, \ldots, C_q$, with $\bigcap_{i=1}^q C_i \neq \emptyset$, let their associated relaxed projection mappings be $T_{C_1}, \ldots, T_{C_q}$. For any $f_0 \in \mathcal{H}$, this defines the sequence of points

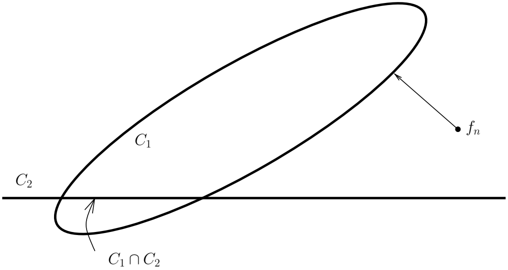
$$f_{n+1} := T_{C_q} \cdots T_{C_1}(f_n), \qquad \forall n.$$



Projections Onto Convex Sets (POCS)

Given a finite number of closed convex sets $C_1, \ldots, C_q$, with $\bigcap_{i=1}^{q} C_i \neq \emptyset$, let their associated relaxed projection mappings be $T_{C_1}, \ldots, T_{C_q}$. For any $f_0 \in \mathcal{H}$, this defines the sequence of points

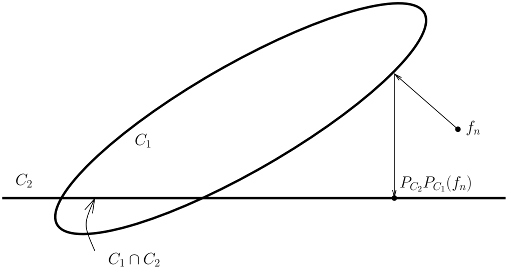
$$f_{n+1} := T_{C_q} \cdots T_{C_1}(f_n), \qquad \forall n.$$



Projections Onto Convex Sets (POCS)

Given a finite number of closed convex sets $C_1, \ldots, C_q$, with $\bigcap_{i=1}^{q} C_i \neq \emptyset$, let their associated relaxed projection mappings be $T_{C_1}, \ldots, T_{C_q}$. For any $f_0 \in \mathcal{H}$, this defines the sequence of points

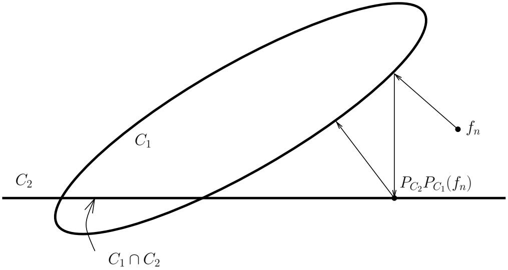
$$f_{n+1} := T_{C_q} \cdots T_{C_1}(f_n), \qquad \forall n.$$



Projections Onto Convex Sets (POCS)

Given a finite number of closed convex sets $C_1, \ldots, C_q$, with $\bigcap_{i=1}^{q} C_i \neq \emptyset$, let their associated relaxed projection mappings be $T_{C_1}, \ldots, T_{C_q}$. For any $f_0 \in \mathcal{H}$, this defines the sequence of points

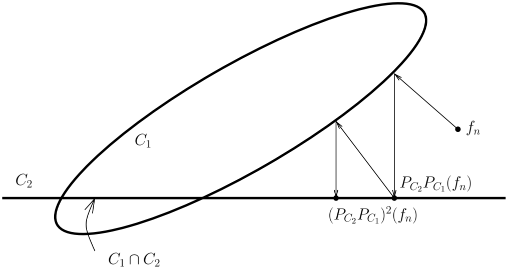
$$f_{n+1} := T_{C_q} \cdots T_{C_1}(f_n), \qquad \forall n.$$



Projections Onto Convex Sets (POCS)

Given a finite number of closed convex sets $C_1, \ldots, C_q$, with $\bigcap_{i=1}^{q} C_i \neq \emptyset$, let their associated relaxed projection mappings be $T_{C_1}, \ldots, T_{C_q}$. For any $f_0 \in \mathcal{H}$, this defines the sequence of points

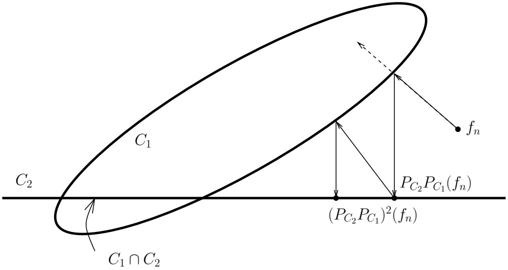
$$f_{n+1} := T_{C_q} \cdots T_{C_1}(f_n), \qquad \forall n.$$



Projections Onto Convex Sets (POCS)

Given a finite number of closed convex sets $C_1, \ldots, C_q$, with $\bigcap_{i=1}^{q} C_i \neq \emptyset$, let their associated relaxed projection mappings be $T_{C_1}, \ldots, T_{C_q}$. For any $f_0 \in \mathcal{H}$, this defines the sequence of points

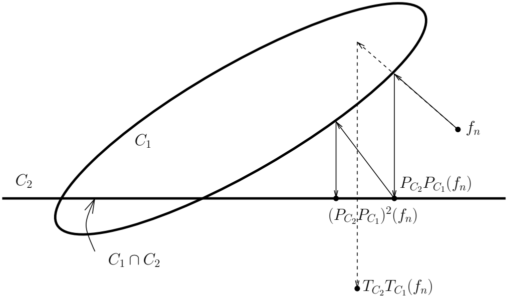
$$f_{n+1} := T_{C_q} \cdots T_{C_1}(f_n), \qquad \forall n.$$



Projections Onto Convex Sets (POCS)

Given a finite number of closed convex sets $C_1, \ldots, C_q$, with $\bigcap_{i=1}^{q} C_i \neq \emptyset$, let their associated relaxed projection mappings be $T_{C_1}, \ldots, T_{C_q}$. For any $f_0 \in \mathcal{H}$, this defines the sequence of points

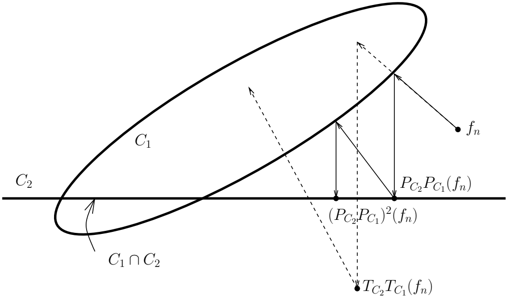
$$f_{n+1} := T_{C_q} \cdots T_{C_1}(f_n), \qquad \forall n.$$



Projections Onto Convex Sets (POCS)

Given a finite number of closed convex sets $C_1, \ldots, C_q$, with $\bigcap_{i=1}^{q} C_i \neq \emptyset$, let their associated relaxed projection mappings be $T_{C_1}, \ldots, T_{C_q}$. For any $f_0 \in \mathcal{H}$, this defines the sequence of points

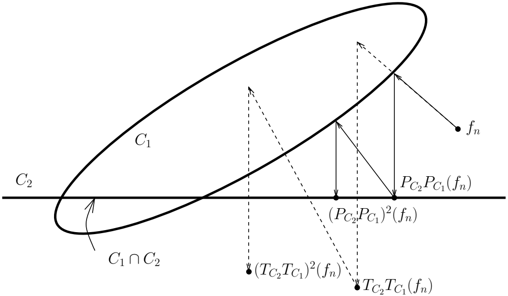
$$f_{n+1} := T_{C_q} \cdots T_{C_1}(f_n), \qquad \forall n.$$



Projections Onto Convex Sets (POCS)

Given a finite number of closed convex sets $C_1, \ldots, C_q$, with $\bigcap_{i=1}^{q} C_i \neq \emptyset$, let their associated relaxed projection mappings be $T_{C_1}, \ldots, T_{C_q}$. For any $f_0 \in \mathcal{H}$, this defines the sequence of points

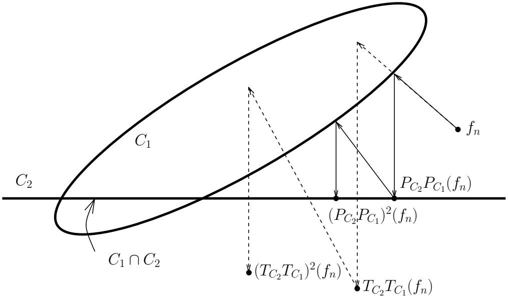
$$f_{n+1} := T_{C_q} \cdots T_{C_1}(f_n), \qquad \forall n.$$



Projections Onto Convex Sets (POCS)

Given a finite number of closed convex sets $C_1, \ldots, C_q$, with $\bigcap_{i=1}^q C_i \neq \emptyset$, let their associated relaxed projection mappings be $T_{C_1}, \ldots, T_{C_q}$. For any $f_0 \in \mathcal{H}$, this defines the sequence of points

$$f_{n+1} := T_{C_q} \cdots T_{C_1}(f_n), \qquad \forall n.$$



Theorem ([Bregman '65], [Gubin, Polyak, Raik '67])

For any $f \in \mathcal{H}$, $\quad (T_{C_q} \cdots T_{C_1})^n(f) \xrightarrow[n\to\infty]{w} \exists f_* \in \bigcap_{i=1}^q C_i$.

Recall

$T_C(f) := f + \mu(P_C(f) - f)$, with $\mu \in (0, 2)$, and $f_{n+1} := T_{C_q} \cdots T_{C_1}(f_n)$, $\forall n$.

Recall

$T_C(f) := f + \mu(P_C(f) - f)$, with $\mu \in (0, 2)$, and $f_{n+1} := T_{C_q} \cdots T_{C_1}(f_n)$, $\forall n$.

Convex Combination of Projection Mappings [Pierra '84]

Given a finite number of closed convex sets $C_1, \ldots, C_q$, with $\bigcap_{i=1}^q C_i \neq \emptyset$, let their associated projection mappings be $P_{C_1}, \ldots, P_{C_q}$. Let also a set of positive constants $w_1, \ldots, w_q$ such that $\sum_{i=1}^q w_i = 1$. Then for any f_0, the sequence

$$f_{n+1} = f_n + \mu_n(\underbrace{\sum_{i=1}^q w_i P_{C_i}(f_n)}_{\text{Convex combination of projections}} - f_n), \quad \forall n,$$

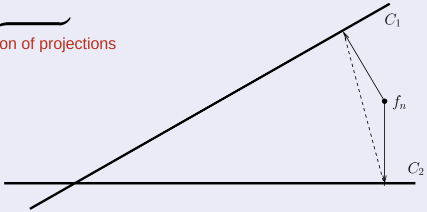
Recall

$T_C(f) := f + \mu(P_C(f) - f)$, with $\mu \in (0,2)$, and $f_{n+1} := T_{C_q} \cdots T_{C_1}(f_n)$, $\forall n$.

Convex Combination of Projection Mappings [Pierra '84]

Given a finite number of closed convex sets $C_1, \ldots, C_q$, with $\bigcap_{i=1}^q C_i \neq \emptyset$, let their associated projection mappings be $P_{C_1}, \ldots, P_{C_q}$. Let also a set of positive constants $w_1, \ldots, w_q$ such that $\sum_{i=1}^q w_i = 1$. Then for any f_0, the sequence

$$f_{n+1} = f_n + \mu_n \Big(\underbrace{\sum_{i=1}^{q} w_i P_{C_i}(f_n)}_{\text{Convex combination of projections}} - f_n \Big), \quad \forall n,$$



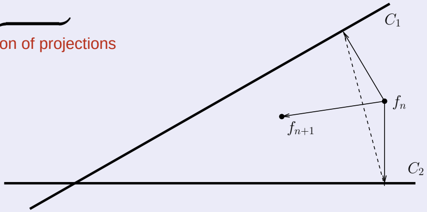
Recall

$T_C(f) := f + \mu(P_C(f) - f)$, with $\mu \in (0,2)$, and $f_{n+1} := T_{C_q} \cdots T_{C_1}(f_n)$, $\forall n$.

Convex Combination of Projection Mappings [Pierra '84]

Given a finite number of closed convex sets $C_1, \ldots, C_q$, with $\bigcap_{i=1}^q C_i \neq \emptyset$, let their associated projection mappings be $P_{C_1}, \ldots, P_{C_q}$. Let also a set of positive constants $w_1, \ldots, w_q$ such that $\sum_{i=1}^q w_i = 1$. Then for any f_0, the sequence

$$f_{n+1} = f_n + \mu_n \Big(\underbrace{\sum_{i=1}^q w_i P_{C_i}(f_n)}_{\text{Convex combination of projections}} - f_n \Big), \quad \forall n,$$



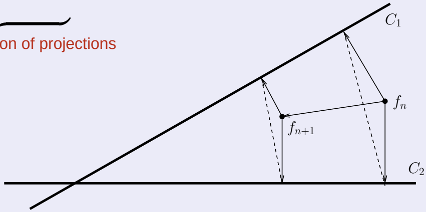
Recall

$T_C(f) := f + \mu(P_C(f) - f)$, with $\mu \in (0, 2)$, and $f_{n+1} := T_{C_q} \cdots T_{C_1}(f_n)$, $\forall n$.

Convex Combination of Projection Mappings [Pierra '84]

Given a finite number of closed convex sets $C_1, \ldots, C_q$, with $\bigcap_{i=1}^q C_i \neq \emptyset$, let their associated projection mappings be $P_{C_1}, \ldots, P_{C_q}$. Let also a set of positive constants $w_1, \ldots, w_q$ such that $\sum_{i=1}^q w_i = 1$. Then for any f_0, the sequence

$$f_{n+1} = f_n + \mu_n\Big(\underbrace{\sum_{i=1}^{q} w_i P_{C_i}(f_n)}_{\text{Convex combination of projections}} - f_n\Big), \quad \forall n,$$



Extrapolated Parallel Projection Method (EPPM)

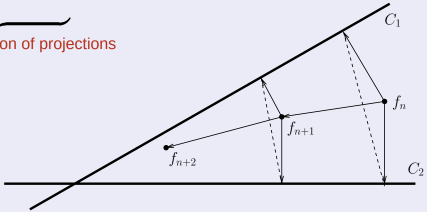
Recall

$T_C(f) := f + \mu(P_C(f) - f)$, with $\mu \in (0, 2)$, and $f_{n+1} := T_{C_q} \cdots T_{C_1}(f_n)$, $\forall n$.

Convex Combination of Projection Mappings [Pierra '84]

Given a finite number of closed convex sets $C_1, \ldots, C_q$, with $\bigcap_{i=1}^{q} C_i \neq \emptyset$, let their associated projection mappings be $P_{C_1}, \ldots, P_{C_q}$. Let also a set of positive constants $w_1, \ldots, w_q$ such that $\sum_{i=1}^{q} w_i = 1$. Then for any f_0, the sequence

$$f_{n+1} = f_n + \mu_n\Big(\underbrace{\sum_{i=1}^{q} w_i P_{C_i}(f_n)}_{\text{Convex combination of projections}} - f_n\Big), \quad \forall n,$$



Extrapolated Parallel Projection Method (EPPM)

Recall

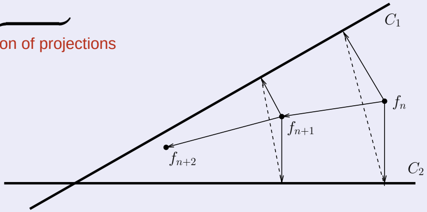
$T_C(f) := f + \mu(P_C(f) - f)$, with $\mu \in (0,2)$, and $f_{n+1} := T_{C_q} \cdots T_{C_1}(f_n)$, $\forall n$.

Convex Combination of Projection Mappings [Pierra '84]

Given a finite number of closed convex sets $C_1, \ldots, C_q$, with $\bigcap_{i=1}^q C_i \neq \emptyset$, let their associated projection mappings be $P_{C_1}, \ldots, P_{C_q}$. Let also a set of positive constants $w_1, \ldots, w_q$ such that $\sum_{i=1}^q w_i = 1$. Then for any f_0, the sequence

$$f_{n+1} = f_n + \mu_n(\underbrace{\sum_{i=1}^{q} w_i P_{C_i}(f_n)}_{\text{Convex combination of projections}} - f_n), \quad \forall n,$$

converges weakly to a point f_* in $\bigcap_{i=1}^q C_i$, where $\mu_n \in (\epsilon, \mathcal{M}_n)$, for $\epsilon \in (0,1)$, and $\mathcal{M}_n := \frac{\sum_{i=1}^q w_i \|P_{C_i}(f_n) - f_n\|^2}{\|\sum_{i=1}^q w_i P_{C_i}(f_n) - f_n\|^2}$.



Infinite Number of Closed Convex Sets

Adaptive Projected Subgradient Method (APSM) [Yamada '03], [Yamada, Ogura '04]

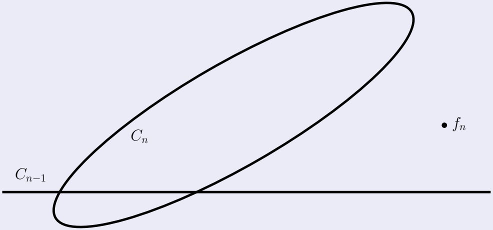
Given an infinite number of closed convex sets $(C_n)_{n\geq 0}$, let their associated projection mappings be (P_{C_n}). For any starting point f_0, let the sequence

$$f_{n+1} = f_n + \mu_n \Big(\sum_{j\in\{n-q+1,\ldots,n\}} w_j P_{C_j}(f_n) - f_n \Big), \quad \forall n,$$

Adaptive Projected Subgradient Method (APSM) [Yamada '03], [Yamada, Ogura '04]

Given an infinite number of closed convex sets $(C_n)_{n\geq 0}$, let their associated projection mappings be (P_{C_n}). For any starting point f_0, let the sequence

$$f_{n+1} = f_n + \mu_n\Big(\sum_{j\in\{n-q+1,\ldots,n\}} w_j P_{C_j}(f_n) - f_n\Big), \quad \forall n,$$

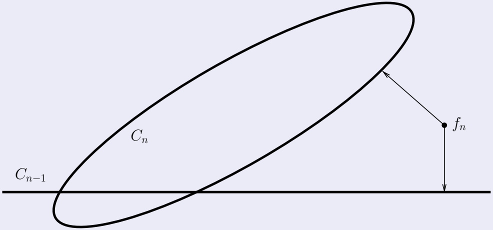


Infinite Number of Closed Convex Sets

Adaptive Projected Subgradient Method (APSM) [Yamada '03], [Yamada, Ogura '04]

Given an infinite number of closed convex sets $(C_n)_{n\geq 0}$, let their associated projection mappings be (P_{C_n}). For any starting point f_0, let the sequence

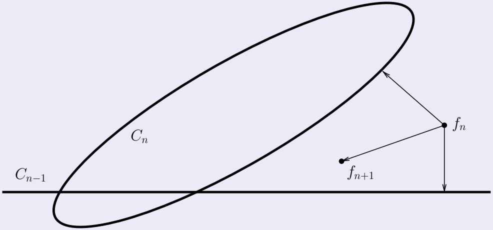
$$f_{n+1} = f_n + \mu_n \Big(\sum_{j\in\{n-q+1,\ldots,n\}} w_j P_{C_j}(f_n) - f_n \Big), \quad \forall n,$$



Adaptive Projected Subgradient Method (APSM) [Yamada '03], [Yamada, Ogura '04]

Given an infinite number of closed convex sets $(C_n)_{n\geq 0}$, let their associated projection mappings be (P_{C_n}). For any starting point f_0, let the sequence

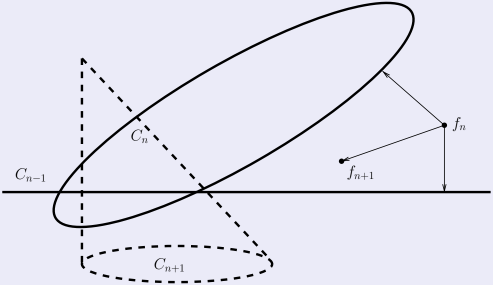
$$f_{n+1} = f_n + \mu_n\Big(\sum_{j\in\{n-q+1,\ldots,n\}} w_j P_{C_j}(f_n) - f_n\Big), \quad \forall n,$$



Adaptive Projected Subgradient Method (APSM) [Yamada '03], [Yamada, Ogura '04]

Given an infinite number of closed convex sets $(C_n)_{n\geq 0}$, let their associated projection mappings be (P_{C_n}). For any starting point f_0, let the sequence

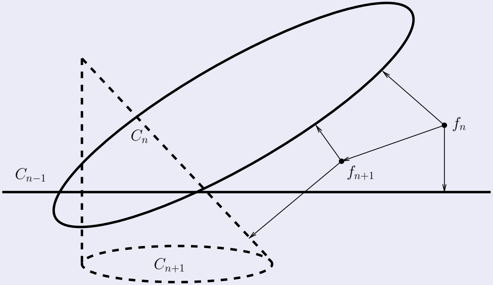
$$f_{n+1} = f_n + \mu_n\Big(\sum_{j\in\{n-q+1,\ldots,n\}} w_j P_{C_j}(f_n) - f_n\Big), \quad \forall n,$$



Adaptive Projected Subgradient Method (APSM) [Yamada '03], [Yamada, Ogura '04]

Given an infinite number of closed convex sets $(C_n)_{n\geq 0}$, let their associated projection mappings be (P_{C_n}). For any starting point f_0, let the sequence

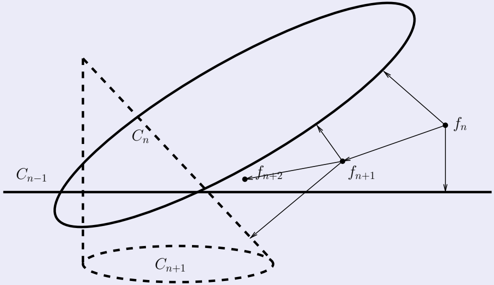
$$f_{n+1} = f_n + \mu_n \Big(\sum_{j\in\{n-q+1,\ldots,n\}} w_j P_{C_j}(f_n) - f_n \Big), \quad \forall n,$$



Adaptive Projected Subgradient Method (APSM) [Yamada '03], [Yamada, Ogura '04]

Given an infinite number of closed convex sets $(C_n)_{n\geq 0}$, let their associated projection mappings be (P_{C_n}). For any starting point f_0, let the sequence

$$f_{n+1} = f_n + \mu_n \Big(\sum_{j\in\{n-q+1,\ldots,n\}} w_j P_{C_j}(f_n) - f_n \Big), \quad \forall n,$$



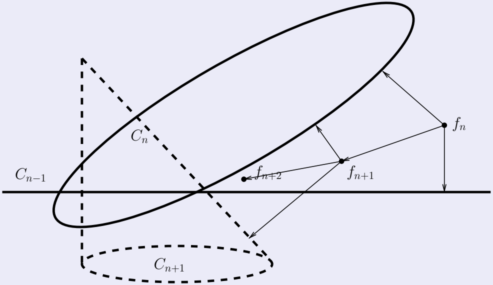
Adaptive Projected Subgradient Method (APSM) [Yamada '03], [Yamada, Ogura '04]

Given an infinite number of closed convex sets $(C_n)_{n\geq 0}$, let their associated projection mappings be (P_{C_n}). For any starting point f_0, let the sequence

$$f_{n+1} = f_n + \mu_n\Big(\sum_{j\in\{n-q+1,\ldots,n\}} w_j P_{C_j}(f_n) - f_n\Big), \quad \forall n,$$

where $\mu_n \in [0, 2\mathcal{M}_n]$, and $\mathcal{M}_n := \frac{\sum_{j\in\{n-q+1,\ldots,n\}} w_j \|P_{C_j}(f_n)-f_n\|^2}{\|\sum_{j\in\{n-q+1,\ldots,n\}} w_j P_{C_j}(f_n)-f_n\|^2}$.

Under certain mild constraints the above sequence converges strongly to a point $f_* \in \mathrm{clos}(\bigcup_{m\geq 0}\bigcap_{n\geq m} C_n)$.



Application to Machine Learning

The Task

Given a set of training samples $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_N \subset \mathbb{R}^m$ and a set of corresponding desired responses $y_0, \ldots, y_N$, estimate a function $f(\cdot) : \mathbb{R}^m \to \mathbb{R}$ that fits the data.

The Task

Given a set of training samples $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_N \subset \mathbb{R}^m$ and a set of corresponding desired responses $y_0, \ldots, y_N$, estimate a function $f(\cdot) : \mathbb{R}^m \to \mathbb{R}$ that fits the data.

The Expected / Empirical Risk Function approach

Estimate f so that the expected risk based on a loss function $\ell(\cdot, \cdot)$ is minimized:

$$\min_f \mathsf{E}\{\ell(f(\boldsymbol{x}), y)\},$$

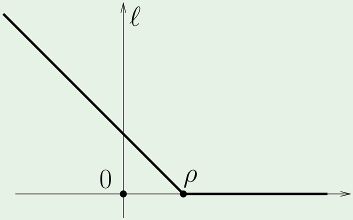
or, in practice, the empirical risk is minimized:

$$\min_f \sum_{n=0}^{N} \ell(f(\boldsymbol{x}_n), y_n).$$

Example (Classification)

For a given margin $\rho \geq 0$, and $y_n \in \{+1, -1\}$, $\forall n$, define the soft margin loss functions:

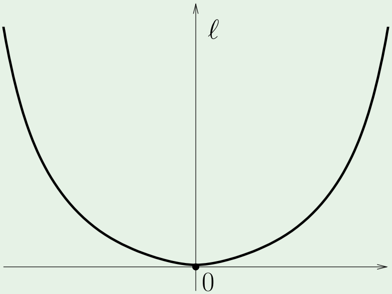
$$\ell(f(\boldsymbol{x}_n), y_n) := \max\{0, \rho - y_n f(\boldsymbol{x}_n)\}, \qquad \forall n.$$



Example (Regression)

The square loss functions:

$$\ell(f(\boldsymbol{x}_n), y_n) := (y_n - f(\boldsymbol{x}_n))^2, \qquad \forall n.$$



Main Idea

The goal here is to have a solution that is in agreement with all the available information, that resides in the data as well as in the available a-priori information.

The Set Theoretic Estimation Approach

Main Idea

The goal here is to have a solution that is in agreement with all the available information, that resides in the data as well as in the available a-priori information.

The Means

- Each piece of information, associated with the training pair $(\boldsymbol{x}_n, y_n)$, is represented in the solution space by a set.

Main Idea

The goal here is to have a solution that is in agreement with all the available information, that resides in the data as well as in the available a-priori information.

The Means

- Each piece of information, associated with the training pair $(\boldsymbol{x}_n, y_n)$, is represented in the solution space by a set.
- Each piece of a-priori information, i.e., each constraint, is also represented by a set.

The Set Theoretic Estimation Approach

Main Idea

The goal here is to have a solution that is in agreement with all the available information, that resides in the data as well as in the available a-priori information.

The Means

- Each piece of information, associated with the training pair $(\boldsymbol{x}_n, y_n)$, is represented in the solution space by a set.
- Each piece of a-priori information, i.e., each constraint, is also represented by a set.
- The intersection of all these sets constitutes the family of solutions.

The Set Theoretic Estimation Approach

Main Idea

The goal here is to have a solution that is in agreement with all the available information, that resides in the data as well as in the available a-priori information.

The Means

- Each piece of information, associated with the training pair $(\boldsymbol{x}_n, y_n)$, is represented in the solution space by a set.
- Each piece of a-priori information, i.e., each constraint, is also represented by a set.
- The intersection of all these sets constitutes the family of solutions.
- The family of solutions is known as the feasibility set.

That is, represent each cost and constraint by an equivalent set C_n and find the solution

$$f \in \bigcap_n C_n \subset \mathcal{H}.$$

Classification: The Soft Margin Loss

The Setting

Let the training data set $(\boldsymbol{x}_n, y_n) \subset \mathbb{R}^m \times \{+1, -1\}$, $n = 0, 1, \ldots$.
Assume the two class task,

$$\begin{cases} y_n = +1, & \boldsymbol{x}_n \in W_1, \\ y_n = -1, & \boldsymbol{x}_n \in W_2. \end{cases}$$

Assume linear separable classes.

Classification: The Soft Margin Loss

The Setting

Let the training data set $(\boldsymbol{x}_n, y_n) \subset \mathbb{R}^m \times \{+1, -1\}$, $n = 0, 1, \ldots$.
Assume the two class task,

$$\begin{cases} y_n = +1, & \boldsymbol{x}_n \in W_1, \\ y_n = -1, & \boldsymbol{x}_n \in W_2. \end{cases}$$

Assume linear separable classes.

The Goal (for $\rho = 0$)

Classification: The Soft Margin Loss

The Setting

Let the training data set $(\boldsymbol{x}_n, y_n) \subset \mathbb{R}^m \times \{+1, -1\}$, $n = 0, 1, \ldots$.
Assume the two class task,

$$\begin{cases} y_n = +1, & \boldsymbol{x}_n \in W_1, \\ y_n = -1, & \boldsymbol{x}_n \in W_2. \end{cases}$$

Assume linear separable classes.

The Goal (for $\rho = 0$)

$$\text{Find} \quad f(\boldsymbol{x}) = \boldsymbol{w}^t \boldsymbol{x} + b, \quad \text{so that}$$

$$\begin{cases} \boldsymbol{w}^t \boldsymbol{x}_n + b \geq 0, & \text{if } y_n = +1, \\ \boldsymbol{w}^t \boldsymbol{x}_n + b \leq 0, & \text{if } y_n = -1. \end{cases}$$

Classification: The Soft Margin Loss

The Setting

Let the training data set $(\boldsymbol{x}_n, y_n) \subset \mathbb{R}^m \times \{+1, -1\}$, $n = 0, 1, \ldots$.
Assume the two class task,

$$\begin{cases} y_n = +1, & \boldsymbol{x}_n \in W_1, \\ y_n = -1, & \boldsymbol{x}_n \in W_2. \end{cases}$$

Assume linear separable classes.

The Goal (for $\rho = 0$)

$$\text{Find} \quad f(\boldsymbol{x}) = \boldsymbol{w}^t \boldsymbol{x} + b, \quad \text{so that}$$

$$\begin{cases} \boldsymbol{w}^t \boldsymbol{x}_n + b \geq 0, & \text{if } y_n = +1, \\ \boldsymbol{w}^t \boldsymbol{x}_n + b \leq 0, & \text{if } y_n = -1. \end{cases}$$

Hereafter, $\left(\boldsymbol{w} \leftarrow \begin{bmatrix} \boldsymbol{w} \\ b \end{bmatrix}, \quad \boldsymbol{x}_n \leftarrow \begin{bmatrix} \boldsymbol{x}_n \\ 1 \end{bmatrix}\right)$.

The Piece of Information

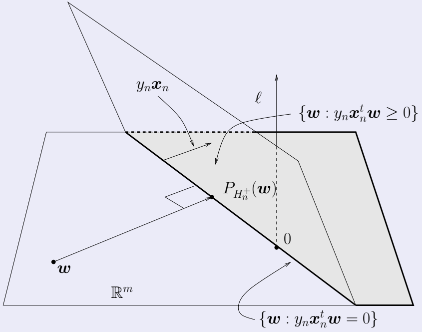
Find all those $\boldsymbol{w}$ so that $\quad y_n \boldsymbol{w}^t \boldsymbol{x}_n \geq 0, \quad n = 0, 1, \ldots$

The Piece of Information

Find all those $\boldsymbol{w}$ so that $\quad y_n \boldsymbol{w}^t \boldsymbol{x}_n \geq 0, \quad n = 0, 1, \ldots$

The Equivalent Set

$$H_n^+ := \{\boldsymbol{w} \in \mathbb{R}^m : y_n \boldsymbol{x}_n^t \boldsymbol{w} \geq 0\}, \; n = 0, 1, \ldots.$$



The feasibility set

For each pair $(\boldsymbol{x}_n, y_n)$, form the equivalent halfspace H_n^+, and

$$\text{find } \boldsymbol{w}_* \in \bigcap_n H_n^+.$$

If linearly separable, the problem is feasible.

The feasibility set

For each pair $(\boldsymbol{x}_n, y_n)$, form the equivalent halfspace H_n^+, and

$$\text{find } \boldsymbol{w}_* \in \bigcap_n H_n^+.$$

If linearly separable, the problem is feasible.

The Algorithm

Each H_n^+ is a convex set.

- Start from an arbitrary initial $\boldsymbol{w}_0$.

The feasibility set

For each pair $(\boldsymbol{x}_n, y_n)$, form the equivalent halfspace H_n^+, and

$$\text{find } \boldsymbol{w}_* \in \bigcap_n H_n^+.$$

If linearly separable, the problem is feasible.

The Algorithm

Each H_n^+ is a convex set.

- Start from an arbitrary initial $\boldsymbol{w}_0$.
- Keep projecting as each H_n^+ is formed.

The feasibility set

For each pair $(\boldsymbol{x}_n, y_n)$, form the equivalent halfspace H_n^+, and

$$\text{find } \boldsymbol{w}_* \in \bigcap_n H_n^+.$$

If linearly separable, the problem is feasible.

The Algorithm

Each H_n^+ is a convex set.

- Start from an arbitrary initial $\boldsymbol{w}_0$.
- Keep projecting as each H_n^+ is formed.
- $P_{H_n^+}(\boldsymbol{w}) = \boldsymbol{w} - \frac{\min\{0, \langle \boldsymbol{w}, y_n \boldsymbol{x}_n \rangle\}}{\|\boldsymbol{x}_n\|^2} y_n \boldsymbol{x}_n$, $\forall \boldsymbol{w} \in \mathcal{H}$.

The feasibility set

For each pair $(\boldsymbol{x}_n, y_n)$, form the equivalent halfspace H_n^+, and

$$\text{find } \boldsymbol{w}_* \in \bigcap_n H_n^+.$$

If linearly separable, the problem is feasible.

The Algorithm

Each H_n^+ is a convex set.

- Start from an arbitrary initial $\boldsymbol{w}_0$.
- Keep projecting as each H_n^+ is formed.
- $P_{H_n^+}(\boldsymbol{w}) = \boldsymbol{w} - \frac{\min\{0, \langle \boldsymbol{w}, y_n \boldsymbol{x}_n \rangle\}}{\|\boldsymbol{x}_n\|^2} y_n \boldsymbol{x}_n$, $\forall \boldsymbol{w} \in \mathcal{H}$.

$\boldsymbol{w}_{n-1}$

The feasibility set

For each pair $(\boldsymbol{x}_n, y_n)$, form the equivalent halfspace H_n^+, and

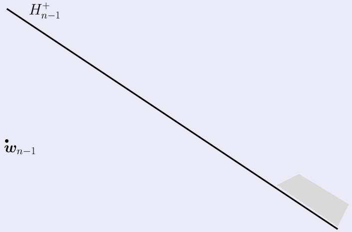
$$\text{find } \boldsymbol{w}_* \in \bigcap_n H_n^+.$$

If linearly separable, the problem is feasible.

The Algorithm

Each H_n^+ is a convex set.

- Start from an arbitrary initial $\boldsymbol{w}_0$.
- Keep projecting as each H_n^+ is formed.
- $P_{H_n^+}(\boldsymbol{w}) = \boldsymbol{w} - \frac{\min\{0, \langle \boldsymbol{w}, y_n \boldsymbol{x}_n \rangle\}}{\|\boldsymbol{x}_n\|^2} y_n \boldsymbol{x}_n$, $\forall \boldsymbol{w} \in \mathcal{H}$.



The feasibility set

For each pair $(\boldsymbol{x}_n, y_n)$, form the equivalent halfspace H_n^+, and

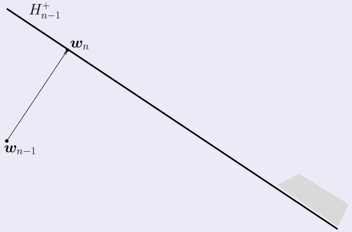
$$\text{find } \boldsymbol{w}_* \in \bigcap_n H_n^+.$$

If linearly separable, the problem is feasible.

The Algorithm

Each H_n^+ is a convex set.

- Start from an arbitrary initial $\boldsymbol{w}_0$.
- Keep projecting as each H_n^+ is formed.
- $P_{H_n^+}(\boldsymbol{w}) = \boldsymbol{w} - \frac{\min\{0, \langle \boldsymbol{w}, y_n \boldsymbol{x}_n \rangle\}}{\|\boldsymbol{x}_n\|^2} y_n \boldsymbol{x}_n$, $\forall \boldsymbol{w} \in \mathcal{H}$.



The feasibility set

For each pair $(\boldsymbol{x}_n, y_n)$, form the equivalent halfspace H_n^+, and

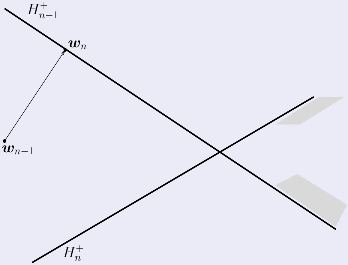
$$\text{find } \boldsymbol{w}_* \in \bigcap_n H_n^+.$$

If linearly separable, the problem is feasible.

The Algorithm

Each H_n^+ is a convex set.

- Start from an arbitrary initial $\boldsymbol{w}_0$.
- Keep projecting as each H_n^+ is formed.
- $P_{H_n^+}(\boldsymbol{w}) = \boldsymbol{w} - \frac{\min\{0, \langle \boldsymbol{w}, y_n \boldsymbol{x}_n \rangle\}}{\|\boldsymbol{x}_n\|^2} y_n \boldsymbol{x}_n$, $\forall \boldsymbol{w} \in \mathcal{H}$.



The feasibility set

For each pair $(\boldsymbol{x}_n, y_n)$, form the equivalent halfspace H_n^+, and

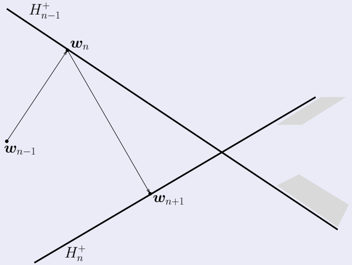
$$\text{find } \boldsymbol{w}_* \in \bigcap_n H_n^+.$$

If linearly separable, the problem is feasible.

The Algorithm

Each H_n^+ is a convex set.

- Start from an arbitrary initial $\boldsymbol{w}_0$.
- Keep projecting as each H_n^+ is formed.
- $P_{H_n^+}(\boldsymbol{w}) = \boldsymbol{w} - \frac{\min\{0, \langle \boldsymbol{w}, y_n \boldsymbol{x}_n \rangle\}}{\|\boldsymbol{x}_n\|^2} y_n \boldsymbol{x}_n$, $\forall \boldsymbol{w} \in \mathcal{H}$.



The feasibility set

For each pair $(\boldsymbol{x}_n, y_n)$, form the equivalent halfspace H_n^+, and

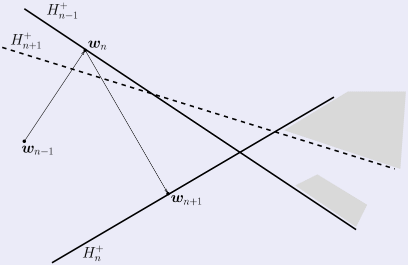
$$\text{find } \boldsymbol{w}_* \in \bigcap_n H_n^+.$$

If linearly separable, the problem is feasible.

The Algorithm

Each H_n^+ is a convex set.

- Start from an arbitrary initial $\boldsymbol{w}_0$.
- Keep projecting as each H_n^+ is formed.
- $P_{H_n^+}(\boldsymbol{w}) = \boldsymbol{w} - \frac{\min\{0, \langle \boldsymbol{w}, y_n \boldsymbol{x}_n \rangle\}}{\|\boldsymbol{x}_n\|^2} y_n \boldsymbol{x}_n$, $\forall \boldsymbol{w} \in \mathcal{H}$.



The feasibility set

For each pair $(\boldsymbol{x}_n, y_n)$, form the equivalent halfspace H_n^+, and

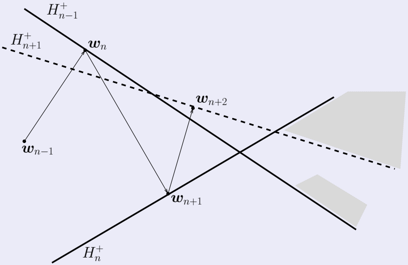
$$\text{find } \boldsymbol{w}_* \in \bigcap_n H_n^+.$$

If linearly separable, the problem is feasible.

The Algorithm

Each H_n^+ is a convex set.

- Start from an arbitrary initial $\boldsymbol{w}_0$.
- Keep projecting as each H_n^+ is formed.
- $P_{H_n^+}(\boldsymbol{w}) = \boldsymbol{w} - \frac{\min\{0, \langle \boldsymbol{w}, y_n \boldsymbol{x}_n \rangle\}}{\|\boldsymbol{x}_n\|^2} y_n \boldsymbol{x}_n$, $\forall \boldsymbol{w} \in \mathcal{H}$.



Algorithmic Solution to Online Classification

$$\boldsymbol{w}_{n+1} := \boldsymbol{w}_n + \mu_n \Big(\sum_{j \in \{n-q+1,\ldots,n\}} \omega_j^{(n)} P_{H_n^+}(\boldsymbol{w}_n) - \boldsymbol{w}_n \Big),$$

$\mu_n \in [0, 2\mathcal{M}_n]$, and

$$\mathcal{M}_n := \begin{cases} \frac{\sum_{j \in \{n-q+1,\ldots,n\}} \omega_j^{(n)} \|P_{H_n^+}(\boldsymbol{w}_n) - \boldsymbol{w}_n\|^2}{\|\sum_{j \in \{n-q+1,\ldots,n\}} \omega_j^{(n)} P_{H_n^+}(\boldsymbol{w}_n) - \boldsymbol{w}_n\|^2}, & \text{if } \boldsymbol{w}_n \notin \bigcap_{j \in \{n-q+1,\ldots,n\}} H_n^+, \\ 1, & \text{otherwise.} \end{cases}$$

$$\boldsymbol{w}_{n+1} := \boldsymbol{w}_n + \mu_n \Big(\sum_{j \in \{n-q+1,\ldots,n\}} \omega_j^{(n)} P_{H_n^+}(\boldsymbol{w}_n) - \boldsymbol{w}_n \Big),$$

$\mu_n \in [0, 2\mathcal{M}_n]$, and

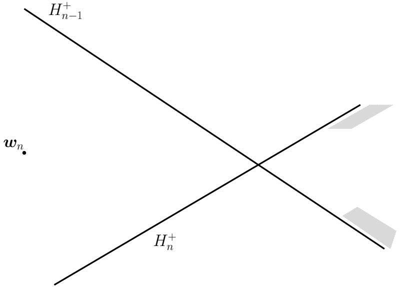
$$\mathcal{M}_n := \begin{cases} \frac{\sum_{j \in \{n-q+1,\ldots,n\}} \omega_j^{(n)} \| P_{H_n^+}(\boldsymbol{w}_n) - \boldsymbol{w}_n \|^2}{\| \sum_{j \in \{n-q+1,\ldots,n\}} \omega_j^{(n)} P_{H_n^+}(\boldsymbol{w}_n) - \boldsymbol{w}_n \|^2}, & \text{if } \boldsymbol{w}_n \notin \bigcap_{j \in \{n-q+1,\ldots,n\}} H_n^+, \\ 1, & \text{otherwise.} \end{cases}$$

$\boldsymbol{w}_n$

$$\boldsymbol{w}_{n+1} := \boldsymbol{w}_n + \mu_n \Big(\sum_{j \in \{n-q+1,\ldots,n\}} \omega_j^{(n)} P_{H_n^+}(\boldsymbol{w}_n) - \boldsymbol{w}_n \Big),$$

$\mu_n \in [0, 2\mathcal{M}_n]$, and

$$\mathcal{M}_n := \begin{cases} \dfrac{\sum_{j\in\{n-q+1,\ldots,n\}} \omega_j^{(n)} \|P_{H_n^+}(\boldsymbol{w}_n) - \boldsymbol{w}_n\|^2}{\|\sum_{j\in\{n-q+1,\ldots,n\}} \omega_j^{(n)} P_{H_n^+}(\boldsymbol{w}_n) - \boldsymbol{w}_n\|^2}, & \text{if } \boldsymbol{w}_n \notin \bigcap_{j\in\{n-q+1,\ldots,n\}} H_n^+, \\ 1, & \text{otherwise.} \end{cases}$$

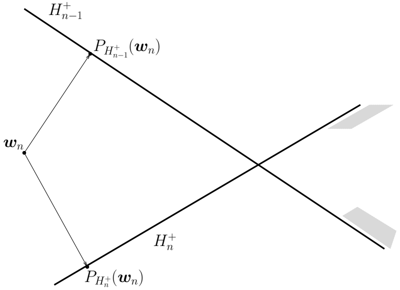


Algorithmic Solution to Online Classification

$$\boldsymbol{w}_{n+1} := \boldsymbol{w}_n + \mu_n \Big(\sum_{j \in \{n-q+1,\ldots,n\}} \omega_j^{(n)} P_{H_n^+}(\boldsymbol{w}_n) - \boldsymbol{w}_n \Big),$$

$\mu_n \in [0, 2\mathcal{M}_n]$, and

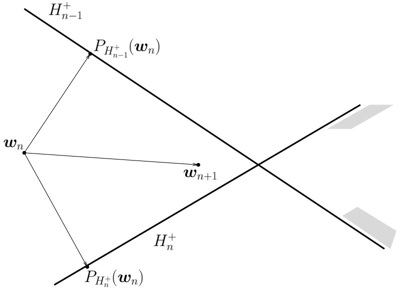
$$\mathcal{M}_n := \begin{cases} \frac{\sum_{j\in\{n-q+1,\ldots,n\}} \omega_j^{(n)} \|P_{H_n^+}(\boldsymbol{w}_n) - \boldsymbol{w}_n\|^2}{\|\sum_{j\in\{n-q+1,\ldots,n\}} \omega_j^{(n)} P_{H_n^+}(\boldsymbol{w}_n) - \boldsymbol{w}_n\|^2}, & \text{if } \boldsymbol{w}_n \notin \bigcap_{j\in\{n-q+1,\ldots,n\}} H_n^+, \\ 1, & \text{otherwise.} \end{cases}$$



$$\boldsymbol{w}_{n+1} := \boldsymbol{w}_n + \mu_n \Big(\sum_{j \in \{n-q+1,\ldots,n\}} \omega_j^{(n)} P_{H_n^+}(\boldsymbol{w}_n) - \boldsymbol{w}_n \Big),$$

$\mu_n \in [0, 2\mathcal{M}_n]$, and

$$\mathcal{M}_n := \begin{cases} \frac{\sum_{j \in \{n-q+1,\ldots,n\}} \omega_j^{(n)} \|P_{H_n^+}(\boldsymbol{w}_n) - \boldsymbol{w}_n\|^2}{\|\sum_{j \in \{n-q+1,\ldots,n\}} \omega_j^{(n)} P_{H_n^+}(\boldsymbol{w}_n) - \boldsymbol{w}_n\|^2}, & \text{if } \boldsymbol{w}_n \notin \bigcap_{j \in \{n-q+1,\ldots,n\}} H_n^+, \\ 1, & \text{otherwise.} \end{cases}$$

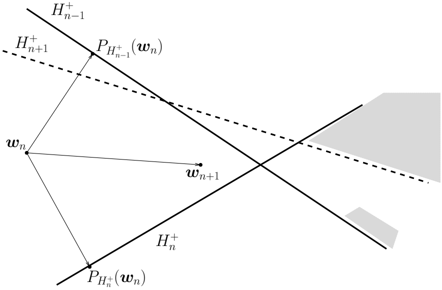


Algorithmic Solution to Online Classification

$$\boldsymbol{w}_{n+1} := \boldsymbol{w}_n + \mu_n \Big(\sum_{j \in \{n-q+1,\ldots,n\}} \omega_j^{(n)} P_{H_n^+}(\boldsymbol{w}_n) - \boldsymbol{w}_n \Big),$$

$\mu_n \in [0, 2\mathcal{M}_n]$, and

$$\mathcal{M}_n := \begin{cases} \dfrac{\sum_{j\in\{n-q+1,\ldots,n\}} \omega_j^{(n)} \|P_{H_n^+}(\boldsymbol{w}_n) - \boldsymbol{w}_n\|^2}{\|\sum_{j\in\{n-q+1,\ldots,n\}} \omega_j^{(n)} P_{H_n^+}(\boldsymbol{w}_n) - \boldsymbol{w}_n\|^2}, & \text{if } \boldsymbol{w}_n \notin \bigcap_{j\in\{n-q+1,\ldots,n\}} H_n^+, \\ 1, & \text{otherwise.} \end{cases}$$

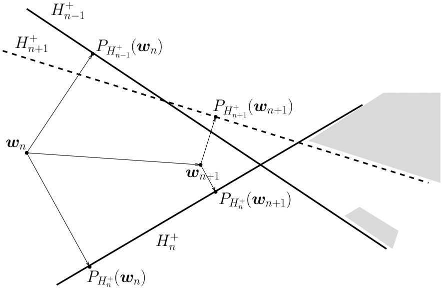


Algorithmic Solution to Online Classification

$$\boldsymbol{w}_{n+1} := \boldsymbol{w}_n + \mu_n\Big(\sum_{j\in\{n-q+1,\dots,n\}} \omega_j^{(n)} P_{H_n^+}(\boldsymbol{w}_n) - \boldsymbol{w}_n\Big),$$

$\mu_n \in [0, 2\mathcal{M}_n]$, and

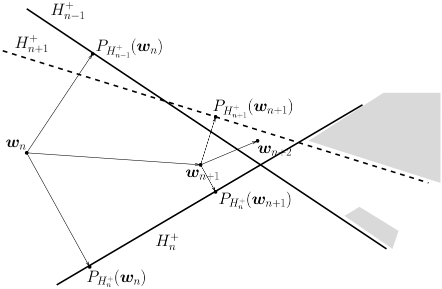
$$\mathcal{M}_n := \begin{cases} \dfrac{\sum_{j\in\{n-q+1,\dots,n\}} \omega_j^{(n)} \|P_{H_n^+}(\boldsymbol{w}_n) - \boldsymbol{w}_n\|^2}{\|\sum_{j\in\{n-q+1,\dots,n\}} \omega_j^{(n)} P_{H_n^+}(\boldsymbol{w}_n) - \boldsymbol{w}_n\|^2}, & \text{if } \boldsymbol{w}_n \notin \bigcap_{j\in\{n-q+1,\dots,n\}} H_n^+, \\ 1, & \text{otherwise.} \end{cases}$$



$$\boldsymbol{w}_{n+1} := \boldsymbol{w}_n + \mu_n \Big(\sum_{j \in \{n-q+1,\ldots,n\}} \omega_j^{(n)} P_{H_n^+}(\boldsymbol{w}_n) - \boldsymbol{w}_n \Big),$$

$\mu_n \in [0, 2\mathcal{M}_n]$, and

$$\mathcal{M}_n := \begin{cases} \dfrac{\sum_{j\in\{n-q+1,\ldots,n\}} \omega_j^{(n)} \| P_{H_n^+}(\boldsymbol{w}_n) - \boldsymbol{w}_n \|^2}{\| \sum_{j\in\{n-q+1,\ldots,n\}} \omega_j^{(n)} P_{H_n^+}(\boldsymbol{w}_n) - \boldsymbol{w}_n \|^2}, & \text{if } \boldsymbol{w}_n \notin \bigcap_{j\in\{n-q+1,\ldots,n\}} H_n^+, \\ 1, & \text{otherwise.} \end{cases}$$



Transfer the Problem into High Dimensional Spaces

Theorem (Cover '65)

The probability of linearly separating any two subgroups of a given finite number of data approaches unity as the dimension of the space, where classification is carried out, increases.

Definition

Consider a Hilbert space $\mathcal{H}$ of functions $f : \mathbb{R}^m \rightarrow \mathbb{R}$.

Definition

Consider a Hilbert space $\mathcal{H}$ of functions $f : \mathbb{R}^m \to \mathbb{R}$.
Assume there exists a kernel function $\kappa(\cdot, \cdot) : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ such that

Definition

Consider a Hilbert space $\mathcal{H}$ of functions $f : \mathbb{R}^m \to \mathbb{R}$.
Assume there exists a kernel function $\kappa(\cdot, \cdot) : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ such that

- $\kappa(\boldsymbol{x}, \cdot) \in \mathcal{H}$, $\forall \boldsymbol{x} \in \mathbb{R}^m$,

Definition

Consider a Hilbert space $\mathcal{H}$ of functions $f : \mathbb{R}^m \rightarrow \mathbb{R}$.
Assume there exists a kernel function $\kappa(\cdot,\cdot) : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}$ such that

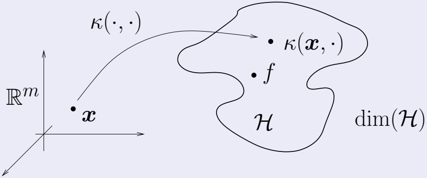
- $\kappa(\boldsymbol{x},\cdot) \in \mathcal{H}$, $\forall \boldsymbol{x} \in \mathbb{R}^m$,
- $\langle f, \kappa(\boldsymbol{x},\cdot)\rangle = f(\boldsymbol{x})$, $\forall \boldsymbol{x} \in \mathbb{R}^m$, $\forall f \in \mathcal{H}$, (reproducing property).

Definition

Consider a Hilbert space $\mathcal{H}$ of functions $f : \mathbb{R}^m \to \mathbb{R}$.
Assume there exists a kernel function $\kappa(\cdot, \cdot) : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ such that

- $\kappa(\boldsymbol{x}, \cdot) \in \mathcal{H}$, $\forall \boldsymbol{x} \in \mathbb{R}^m$,
- $\langle f, \kappa(\boldsymbol{x}, \cdot)\rangle = f(\boldsymbol{x})$, $\forall \boldsymbol{x} \in \mathbb{R}^m$, $\forall f \in \mathcal{H}$, (reproducing property).

Then $\mathcal{H}$ is called a Reproducing Kernel Hilbert Space (RKHS).



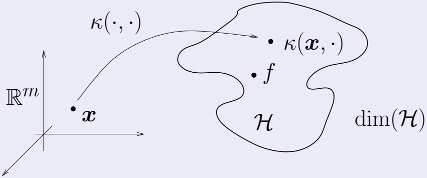
Reproducing Kernel Hilbert Spaces (RKHS)

Definition

Consider a Hilbert space $\mathcal{H}$ of functions $f : \mathbb{R}^m \to \mathbb{R}$.
Assume there exists a kernel function $\kappa(\cdot,\cdot) : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ such that

- $\kappa(\boldsymbol{x},\cdot) \in \mathcal{H}$, $\forall \boldsymbol{x} \in \mathbb{R}^m$,
- $\langle f, \kappa(\boldsymbol{x},\cdot)\rangle = f(\boldsymbol{x})$, $\forall \boldsymbol{x} \in \mathbb{R}^m$, $\forall f \in \mathcal{H}$, (reproducing property).

Then $\mathcal{H}$ is called a Reproducing Kernel Hilbert Space (RKHS).



Properties

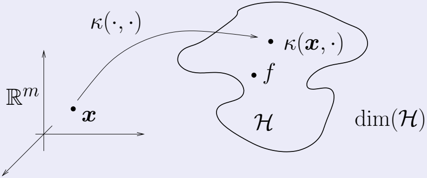
Reproducing Kernel Hilbert Spaces (RKHS)

Definition

Consider a Hilbert space $\mathcal{H}$ of functions $f : \mathbb{R}^m \to \mathbb{R}$.
Assume there exists a kernel function $\kappa(\cdot, \cdot) : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ such that

- $\kappa(\boldsymbol{x}, \cdot) \in \mathcal{H}$, $\forall \boldsymbol{x} \in \mathbb{R}^m$,
- $\langle f, \kappa(\boldsymbol{x}, \cdot)\rangle = f(\boldsymbol{x})$, $\forall \boldsymbol{x} \in \mathbb{R}^m$, $\forall f \in \mathcal{H}$, (reproducing property).

Then $\mathcal{H}$ is called a Reproducing Kernel Hilbert Space (RKHS).



Properties

- Kernel Trick: $\langle \kappa(\boldsymbol{x}, \cdot), \kappa(\boldsymbol{y}, \cdot)\rangle = \kappa(\boldsymbol{x}, \boldsymbol{y})$.

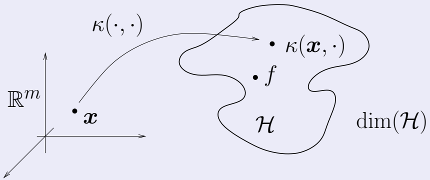
Reproducing Kernel Hilbert Spaces (RKHS)

Definition

Consider a Hilbert space $\mathcal{H}$ of functions $f : \mathbb{R}^m \to \mathbb{R}$.
Assume there exists a kernel function $\kappa(\cdot,\cdot) : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ such that

- $\kappa(\boldsymbol{x},\cdot) \in \mathcal{H}$, $\forall \boldsymbol{x} \in \mathbb{R}^m$,
- $\langle f, \kappa(\boldsymbol{x},\cdot)\rangle = f(\boldsymbol{x})$, $\forall \boldsymbol{x} \in \mathbb{R}^m$, $\forall f \in \mathcal{H}$, (reproducing property).

Then $\mathcal{H}$ is called a Reproducing Kernel Hilbert Space (RKHS).



Properties

- Kernel Trick: $\langle \kappa(\boldsymbol{x},\cdot), \kappa(\boldsymbol{y},\cdot)\rangle = \kappa(\boldsymbol{x},\boldsymbol{y})$.
- $\mathcal{H} = \operatorname{clos}\{\sum_{n=0}^{N} \gamma_n \kappa(\boldsymbol{x}_n,\cdot) : \forall \boldsymbol{x}_n \in \mathbb{R}^m, \forall \gamma_n, \forall N\}$.

The Goal

Let the training data set $(\boldsymbol{x}_n, y_n) \subset \mathbb{R}^m \times \{+1, -1\}$, $n = 0, 1, \ldots$.

- $\boldsymbol{x}_n \mapsto \kappa(\boldsymbol{x}_n, \cdot)$,

Classification in RKHS

The Goal

Let the training data set $(\boldsymbol{x}_n, y_n) \subset \mathbb{R}^m \times \{+1, -1\}$, $n = 0, 1, \ldots$.

- $\boldsymbol{x}_n \mapsto \kappa(\boldsymbol{x}_n, \cdot)$,
- Find $f \in \mathcal{H}$ and $b \in \mathbb{R}$ so that

$$y_n(f(\boldsymbol{x}_n) + b) = y_n(\langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle + b) \geq 0, \qquad \forall n.$$

The Piece of Information

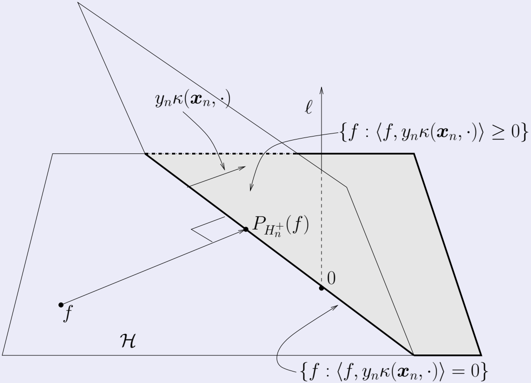
Find all those f so that $\langle f, y_n \kappa(\boldsymbol{x}_n, \cdot)\rangle \geq 0, \quad n = 0, 1, \ldots$

The Piece of Information

Find all those f so that $\langle f, y_n\kappa(\boldsymbol{x}_n,\cdot)\rangle \geq 0, \quad n = 0, 1, \ldots$

The Equivalence Set

$$H_n^+ := \{f \in \mathcal{H} : \langle f, y_n\kappa(\boldsymbol{x}_n,\cdot)\rangle \geq 0\},\ n = 0, 1, \ldots.$$



Let the index set $\mathcal{J}_n := \{n-q+1, \ldots, n\}$. Also the weights $\omega_j^{(n)} \geq 0$ such that $\sum_{j\in\mathcal{J}_n} \omega_j^{(n)} = 1$. For $f_0 \in \mathcal{H}$,

$$f_{n+1} := f_n + \mu_n \Big(\sum_{j\in\mathcal{J}_n} \omega_j^{(n)} P_{H_j^+}(f_n) - f_n \Big), \quad \forall n \geq 0,$$

where the extrapolation coefficient $\mu_n \in [0, 2\mathcal{M}_n]$ with

$$\mathcal{M}_n := \begin{cases} \frac{\sum_{j\in\mathcal{J}_n} \omega_j^{(n)} \|P_{H_j^+}(f_n) - f_n\|^2}{\|\sum_{j\in\mathcal{J}_n} \omega_j^{(n)} P_{H_j^+}(f_n) - f_n\|^2}, & \text{if } f_n \notin \bigcap_{j\in\mathcal{J}_n} H_j^+, \\ 1, & \text{otherwise.} \end{cases}$$

Representer Theorem

Theorem

By mathematical induction on the previous algorithmic procedure, for each index n, there exist $(\gamma_i^{(n)})$ such that

$$f_n := \sum_{i=0}^{n-1} \gamma_i^{(n)} \kappa(\boldsymbol{x}_i, \cdot).$$

Sparsification

Recall that as time goes by:

$$f_n := \sum_{i=0}^{n-1} \gamma_i^{(n)} \kappa(\boldsymbol{x}_i, \cdot).$$

Sparsification

Recall that as time goes by:

$$f_n := \sum_{i=0}^{n-1} \gamma_i^{(n)} \kappa(\boldsymbol{x}_i, \cdot).$$

Memory and computational load grows unbounded as $n \to \infty$!

Sparsification

Recall that as time goes by:

$$f_n := \sum_{i=0}^{n-1} \gamma_i^{(n)} \kappa(\boldsymbol{x}_i, \cdot).$$

Memory and computational load grows unbounded as $n \to \infty$!

To cope with the problem, we additionally constrain the norm of f_n by a predefined $\delta > 0$ [Slavakis, Theodoridis, Yamada '08]:

$$(\forall n \geq 0) \ f_n \in \mathcal{B} := \{f \in \mathcal{H} : \|f\| \leq \delta\} : \text{ Closed Ball.}$$

Sparsification

Recall that as time goes by:

$$f_n := \sum_{i=0}^{n-1} \gamma_i^{(n)} \kappa(\boldsymbol{x}_i, \cdot).$$

Memory and computational load grows unbounded as $n \to \infty$!

To cope with the problem, we additionally constrain the norm of f_n by a predefined $\delta > 0$ [Slavakis, Theodoridis, Yamada '08]:

$$(\forall n \geq 0)\ f_n \in \mathcal{B} := \{f \in \mathcal{H} : \|f\| \leq \delta\} : \text{ Closed Ball.}$$

Goal

Thus, we are looking for a classifier $f \in \mathcal{H}$ such that

$$f \in \mathcal{B} \cap \Big(\bigcap_n H_n^+\Big).$$

$$f_{n+1} := P_{\mathcal{B}}\left(f_n + \mu_n\big(\sum_{j\in\mathcal{J}_n}\omega_j^{(n)} P_{H_j^+}(f_n) - f_n\big)\right), \qquad \forall n \in \mathbb{Z}_{\geq 0}.$$

$$\mu_n \in [0, 2\mathcal{M}_n], \quad \mathcal{M}_n \geq 1,$$

Geometric Illustration of the Algorithm

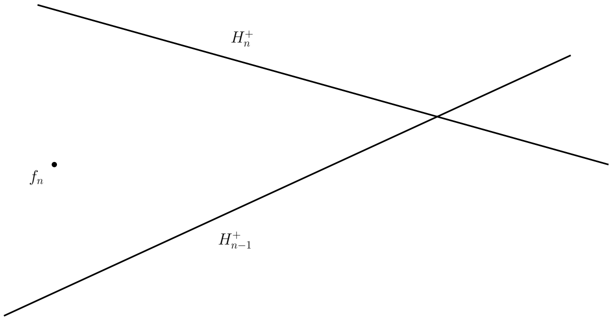
$$f_{n+1} := P_{\mathcal{B}} \left(f_n + \mu_n \Big(\sum_{j \in \mathcal{J}_n} \omega_j^{(n)} P_{H_j^+}(f_n) - f_n \Big) \right), \qquad \forall n \in \mathbb{Z}_{\geq 0}.$$

$$\mu_n \in [0, 2\mathcal{M}_n], \quad \mathcal{M}_n \geq 1,$$

Geometric Illustration of the Algorithm

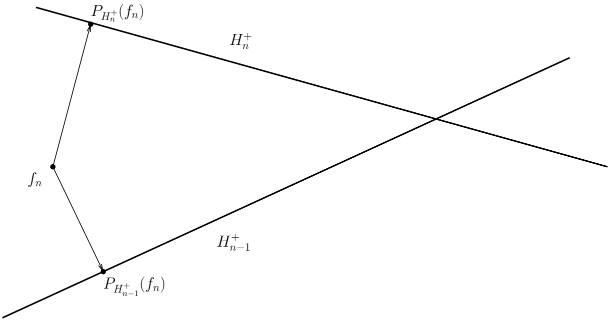
$$f_{n+1} := P_{\mathcal{B}}\left(f_n + \mu_n \Big(\sum_{j \in \mathcal{J}_n} \omega_j^{(n)} P_{H_j^+}(f_n) - f_n \Big) \right), \qquad \forall n \in \mathbb{Z}_{\geq 0}.$$

$$\mu_n \in [0, 2\mathcal{M}_n], \quad \mathcal{M}_n \geq 1,$$



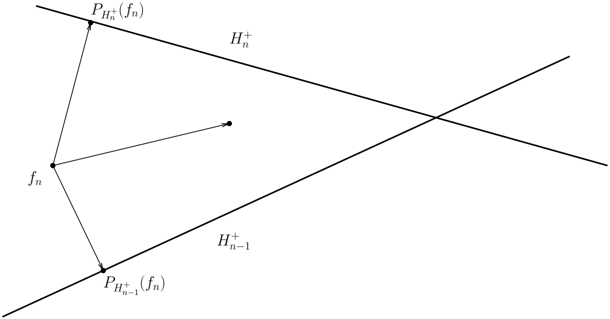
$$f_{n+1} := P_{\mathcal{B}}\left(f_n + \mu_n \Big(\sum_{j \in \mathcal{J}_n} \omega_j^{(n)} P_{H_j^+}(f_n) - f_n \Big) \right), \qquad \forall n \in \mathbb{Z}_{\geq 0}.$$

$$\mu_n \in [0, 2\mathcal{M}_n], \quad \mathcal{M}_n \geq 1,$$



$$f_{n+1} := P_{\mathcal{B}}\left(f_n + \mu_n \Big(\sum_{j \in \mathcal{J}_n} \omega_j^{(n)} P_{H_j^+}(f_n) - f_n \Big) \right), \qquad \forall n \in \mathbb{Z}_{\geq 0}.$$

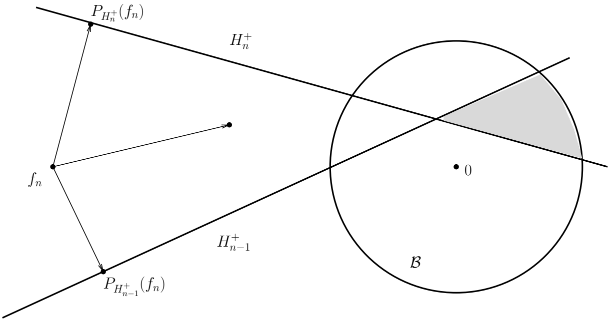
$$\mu_n \in [0, 2\mathcal{M}_n], \quad \mathcal{M}_n \geq 1,$$



Geometric Illustration of the Algorithm

$$f_{n+1} := P_{\mathcal{B}}\left(f_n + \mu_n\Big(\sum_{j\in\mathcal{J}_n}\omega_j^{(n)} P_{H_j^+}(f_n) - f_n\Big)\right), \qquad \forall n \in \mathbb{Z}_{\geq 0}.$$

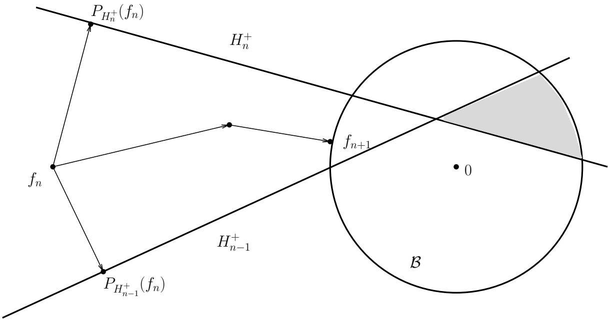
$$\mu_n \in [0, 2\mathcal{M}_n], \quad \mathcal{M}_n \geq 1,$$



Geometric Illustration of the Algorithm

$$f_{n+1} := P_{\mathcal{B}}\left(f_n + \mu_n \Big(\sum_{j\in\mathcal{J}_n} \omega_j^{(n)} P_{H_j^+}(f_n) - f_n \Big) \right), \qquad \forall n \in \mathbb{Z}_{\geq 0}.$$

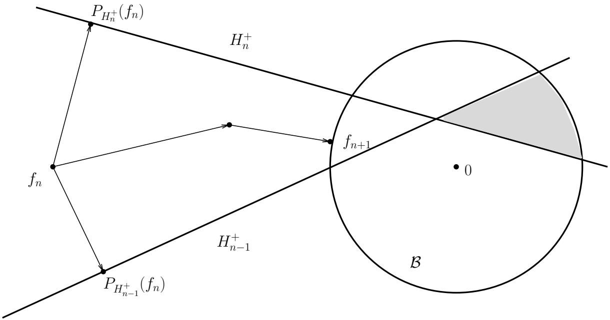
$$\mu_n \in [0, 2\mathcal{M}_n], \quad \mathcal{M}_n \geq 1,$$



Geometric Illustration of the Algorithm

$$f_{n+1} := P_{\mathcal{B}}\left(f_n + \mu_n \Big(\sum_{j \in \mathcal{J}_n} \omega_j^{(n)} P_{H_j^+}(f_n) - f_n \Big) \right), \qquad \forall n \in \mathbb{Z}_{\geq 0}.$$

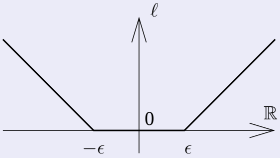
$$\mu_n \in [0, 2\mathcal{M}_n], \quad \mathcal{M}_n \geq 1,$$



Remark: It can be shown that this scheme leads to a forgetting factor effect, as in adaptive filtering!

The linear ϵ-insensitive loss function case

$$\ell(x) := \max\{0, |x| - \epsilon\}, x \in \mathbb{R}.$$



The Piece of Information

Given $(\boldsymbol{x}_n, y_n) \in \mathbb{R}^m \times \mathbb{R}$, find $f \in \mathcal{H}$ such that

$$|\langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - y_n| \leq \epsilon, \qquad \forall n.$$

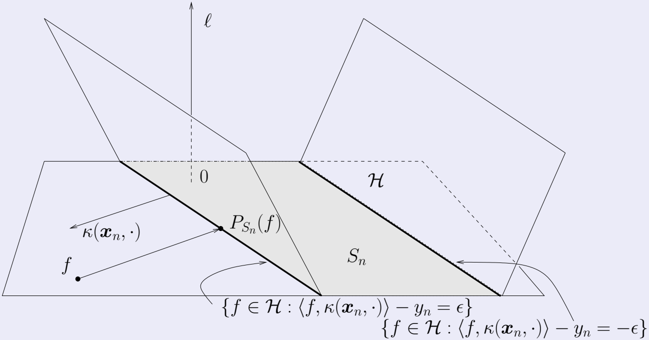
The Piece of Information

Given $(\boldsymbol{x}_n, y_n) \in \mathbb{R}^m \times \mathbb{R}$, find $f \in \mathcal{H}$ such that

$$|\langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - y_n| \leq \epsilon, \qquad \forall n.$$

The Equivalence Set (Hyperslab)

$$S_n := \{f \in \mathcal{H} : |\langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - y_n| \leq \epsilon\}, \quad \forall n.$$



Projection onto a Hyperslab

$$P_{S_n}(f) = f + \beta\kappa(\boldsymbol{x}_n, \cdot), \forall f \in \mathcal{H},$$

where

$$\beta := \begin{cases} \frac{y_n - \langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - \epsilon}{\kappa(\boldsymbol{x}_n, \boldsymbol{x}_n)}, & \text{if } \langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - y_n < -\epsilon, \\ 0, & \text{if } |\langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - y_n| \leq \epsilon, \\ -\frac{\langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - y_n - \epsilon}{\kappa(\boldsymbol{x}_n, \boldsymbol{x}_n)}, & \text{if } \langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - y_n > \epsilon. \end{cases}$$

Projection onto a Hyperslab

$$P_{S_n}(f) = f + \beta\kappa(\boldsymbol{x}_n, \cdot), \forall f \in \mathcal{H},$$

where

$$\beta := \begin{cases} \frac{y_n - \langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - \epsilon}{\kappa(\boldsymbol{x}_n, \boldsymbol{x}_n)}, & \text{if } \langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - y_n < -\epsilon, \\ 0, & \text{if } |\langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - y_n| \leq \epsilon, \\ -\frac{\langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - y_n - \epsilon}{\kappa(\boldsymbol{x}_n, \boldsymbol{x}_n)}, & \text{if } \langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - y_n > \epsilon. \end{cases}$$

The feasibility set

For each pair $(\boldsymbol{x}_n, y_n)$, form the equivalent hyperslab S_n, and

$$\text{find } f_* \in \bigcap_n S_n.$$

Let the index set $\mathcal{J}_n := \{n-q+1,\dots,n\}$. Also the weights $\omega_j^{(n)} \geq 0$ such that $\sum_{j\in\mathcal{J}_n}\omega_j^{(n)} = 1$. For $f_0 \in \mathcal{H}$,

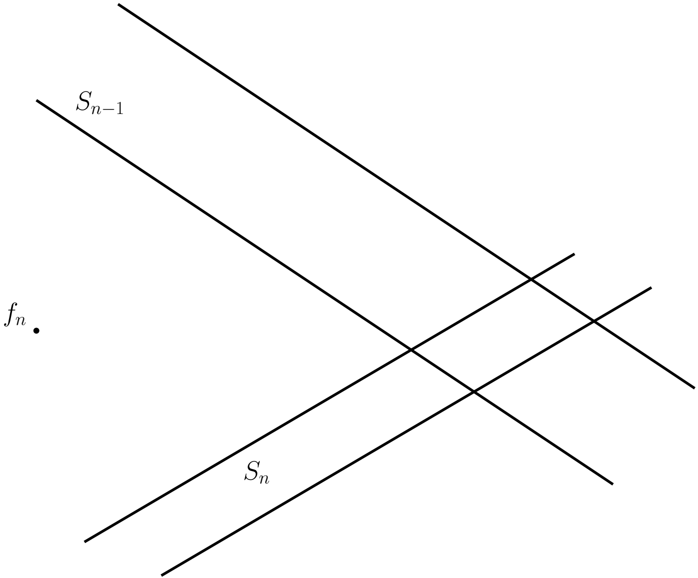
$$f_{n+1} := f_n + \mu_n\Big(\sum_{j\in\mathcal{J}_n}\omega_j^{(n)} P_{S_j}(f_n) - f_n\Big), \quad \forall n \geq 0,$$

where the extrapolation coefficient $\mu_n \in [0, 2\mathcal{M}_n]$ with

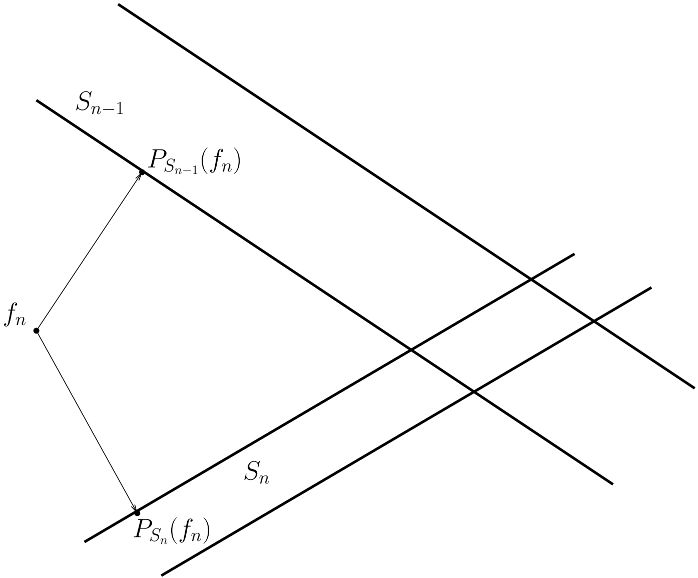
$$\mathcal{M}_n := \begin{cases} \frac{\sum_{j\in\mathcal{J}_n}\omega_j^{(n)}\|P_{S_j}(f_n)-f_n\|^2}{\|\sum_{j\in\mathcal{J}_n}\omega_j^{(n)}P_{S_j}(f_n)-f_n\|^2}, & \text{if } f_n \notin \bigcap_{j\in\mathcal{J}_n} S_j, \\ 1, & \text{otherwise.} \end{cases}$$

Geometric Illustration of the Algorithm

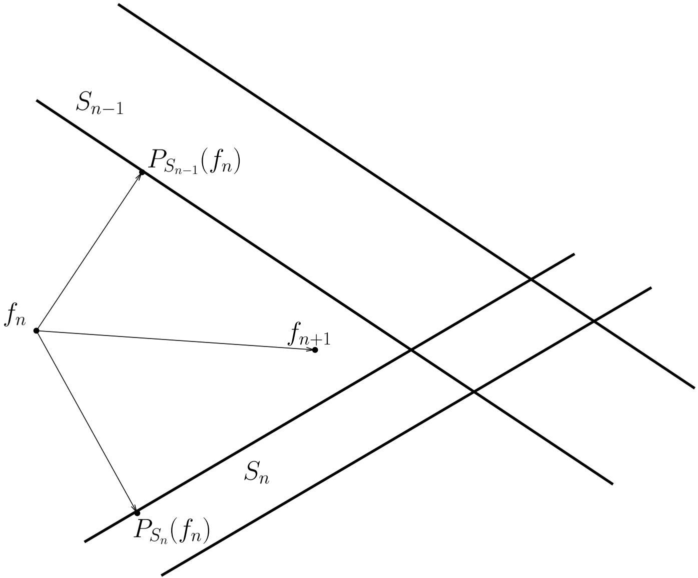
Geometric Illustration of the Algorithm



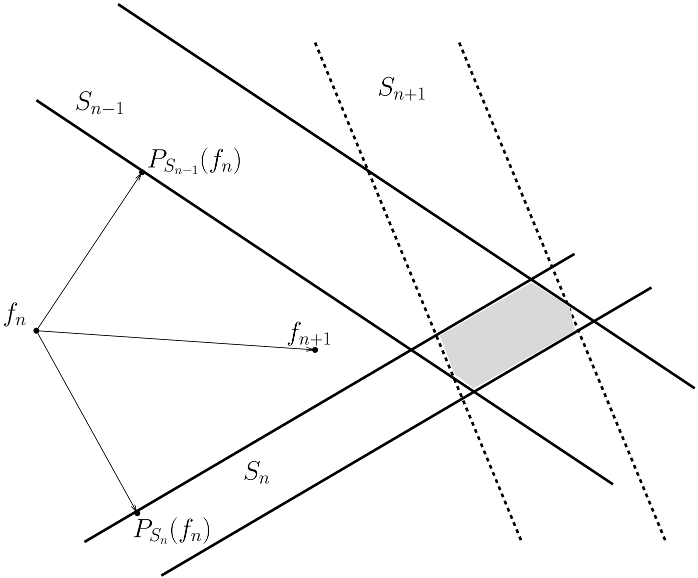
Geometric Illustration of the Algorithm

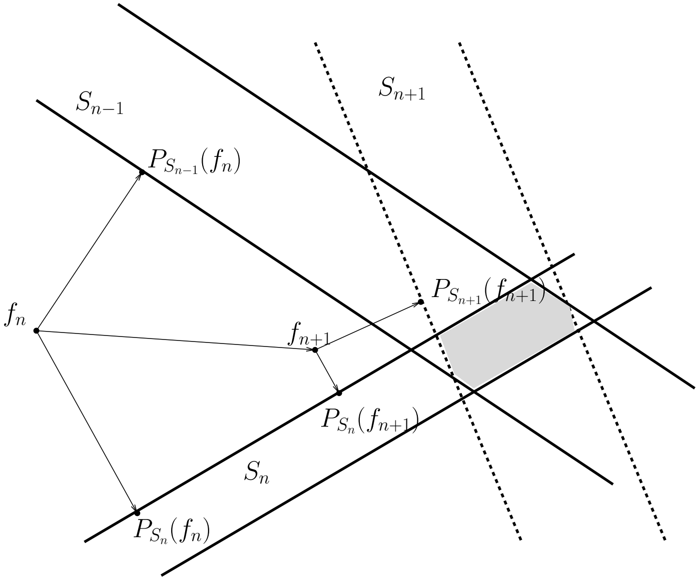


Geometric Illustration of the Algorithm

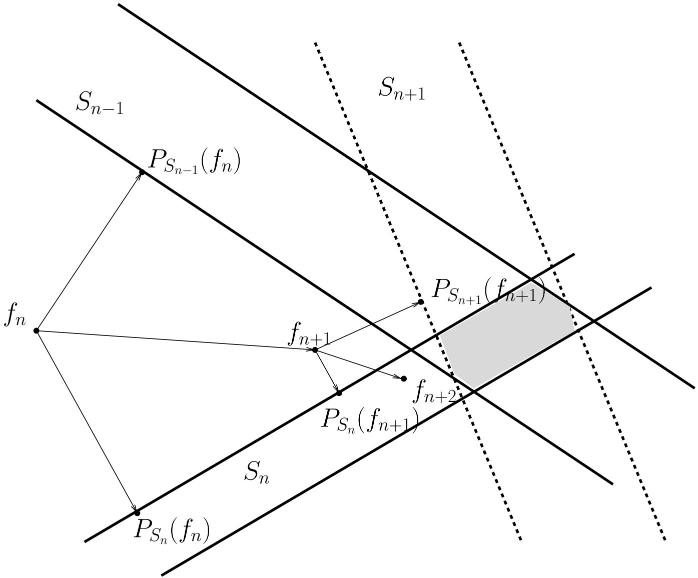


Geometric Illustration of the Algorithm



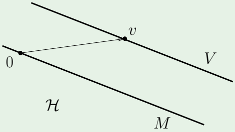

S_{n-1}
S_{n+1}
$P_{S_{n-1}}(f_n)$
$P_{S_{n+1}}(f_{n+1})$
f_n
f_{n+1}
$P_{S_n}(f_{n+1})$
S_n
$P_{S_n}(f_n)$

Geometric Illustration of the Algorithm



Example (Affine Set)

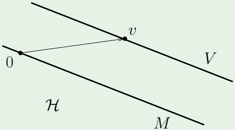
An affine set V is the translation of a closed subspace M, i.e., $V := v + M$, where $v \in V$.



$$P_V(f) = v + P_M(f - v), \forall f \in \mathcal{H}.$$

Example (Affine Set)

An affine set V is the translation of a closed subspace M, i.e., $V := v + M$, where $v \in V$.



$$P_V(f) = v + P_M(f - v), \forall f \in \mathcal{H}.$$

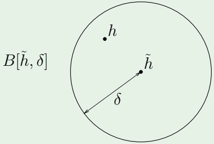
For example, if $M = \text{span}\{\tilde{h}_1, \ldots, \tilde{h}_p\}$, then

$$P_V(f) = v + [\tilde{h}_1, \ldots, \tilde{h}_p]\boldsymbol{G}^\dagger \begin{bmatrix} \langle f-v, \tilde{h}_1 \rangle \\ \vdots \\ \langle f-v, \tilde{h}_p \rangle \end{bmatrix}, \quad \forall f \in \mathcal{H},$$

where the $p \times p$ matrix $\boldsymbol{G}$, with $\boldsymbol{G}_{ij} := \langle \tilde{h}_i, \tilde{h}_j \rangle$, is a Gram matrix, and $\boldsymbol{G}^\dagger$ is the Moore-Penrose pseudoinverse of $\boldsymbol{G}$. The notation $[\tilde{h}_1, \ldots, \tilde{h}_p]\boldsymbol{\gamma} := \sum_{i=1}^p \gamma_i \tilde{h}_i$, for any p-dimensional vector $\boldsymbol{\gamma}$.

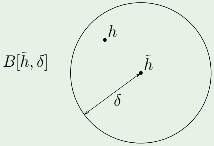
Example (Icecream Cone)

Find $f \in \mathcal{H}$ such that $\langle f, h \rangle \geq \gamma, \ \forall h \in B[\tilde{h}, \delta]$:
(Robustness is desired).



Example (Icecream Cone)

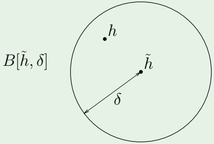
Find $f \in \mathcal{H}$ such that $\langle f, h \rangle \geq \gamma, \ \forall h \in B[\tilde{h}, \delta]$: (Robustness is desired).



If Γ is the set of all such solutions, then

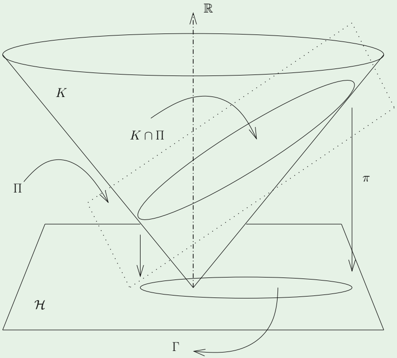
Example (Icecream Cone)

Find $f \in \mathcal{H}$ such that $\langle f, h \rangle \geq \gamma, \ \forall h \in B[\tilde{h}, \delta]$:
(Robustness is desired).



If Γ is the set of all such solutions, then

Find a point in $K \cap \Pi$,
K: an icecream cone,
Π: a hyperplane.



The Complete Picture

Given $(\boldsymbol{x}_n, y_n)$, find an $f \in \mathcal{H}$ such that [Slavakis, Theodoridis '07 and '08]

$$|\langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - y_n| \leq \epsilon \qquad \text{subject to}$$

Given $(\boldsymbol{x}_n, y_n)$, find an $f \in \mathcal{H}$ such that [Slavakis, Theodoridis '07 and '08]

$$|\langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - y_n| \leq \epsilon \qquad \text{subject to}$$

$$f \in V \quad \text{(Affine constraint)}, \qquad \text{and / or}$$

The Complete Picture

Given $(\boldsymbol{x}_n, y_n)$, find an $f \in \mathcal{H}$ such that [Slavakis, Theodoridis '07 and '08]

$$|\langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - y_n| \leq \epsilon \qquad \text{subject to}$$

$$f \in V \quad \text{(Affine constraint),} \qquad \text{and / or}$$

$$\langle f, h\rangle \geq \gamma,\ \forall h \in B[\tilde{h}, \delta] \qquad \text{(Robustness).}$$

Let the index set $\mathcal{J}_n := \{n-q+1, \ldots, n\}$. Also the weights $\omega_j^{(n)} \geq 0$ such that $\sum_{j \in \mathcal{J}_n} \omega_j^{(n)} = 1$. For $f_0 \in \mathcal{H}$,

$$f_{n+1} := P_{\Pi} P_K \left(f_n + \mu_n (\sum_{j \in \mathcal{J}_n} \omega_j^{(n)} P_{S_j}(f_n) - f_n) \right), \quad \forall n \geq 0,$$

where the extrapolation coefficient $\mu_n \in [0, 2\mathcal{M}_n]$ with

$$\mathcal{M}_n := \begin{cases} \frac{\sum_{j \in \mathcal{J}_n} \omega_j^{(n)} \|P_{S_j}(f_n) - f_n\|^2}{\|\sum_{j \in \mathcal{J}_n} \omega_j^{(n)} P_{S_j}(f_n) - f_n\|^2}, & \text{if } f_n \notin \bigcap_{j \in \mathcal{J}_n} S_j, \\ 1, & \text{otherwise.} \end{cases}$$

Theorem

By mathematical induction on the previous algorithmic procedure, for each index n, there exist $(\gamma_i^{(n)})$, and $(\alpha_i^{(n)})$ such that [Slavakis, Theodoridis '08]

$$f_n := \underbrace{\sum_{l=1}^{L_c} \alpha_l^{(n)} \tilde{h}_l}_{\text{Constraints}} + \underbrace{\sum_{i=0}^{n-1} \gamma_i^{(n)} \kappa(\boldsymbol{x}_i, \cdot)}_{\text{Training Data}}, \qquad \forall n.$$

Sparsification

Recall that

$$f_n := \sum_{l=1}^{L_c} \alpha_l^{(n)} \tilde{h}_l + \sum_{i=0}^{n-1} \gamma_i^{(n)} \kappa(\boldsymbol{x}_i, \cdot), \qquad \forall n.$$

Sparsification

Recall that

$$f_n := \sum_{l=1}^{L_c} \alpha_l^{(n)} \tilde{h}_l + \sum_{i=0}^{n-1} \gamma_i^{(n)} \kappa(\boldsymbol{x}_i, \cdot), \qquad \forall n.$$

Memory and computational load grows unbounded as $n \to \infty$!

Recall that

$$f_n := \sum_{l=1}^{L_c} \alpha_l^{(n)} \tilde{h}_l + \sum_{i=0}^{n-1} \gamma_i^{(n)} \kappa(\boldsymbol{x}_i, \cdot), \qquad \forall n.$$

Memory and computational load grows unbounded as $n \to \infty$!

Additionally constrain the norm of f_n by a predefined $\delta > 0$:

$$(\forall n \geq 0)\ f_n \in \mathcal{B} := \{f \in \mathcal{H} : \|f\| \leq \delta\} :$$ Closed Ball.

Recall that

$$f_n := \sum_{l=1}^{L_c} \alpha_l^{(n)} \tilde{h}_l + \sum_{i=0}^{n-1} \gamma_i^{(n)} \kappa(\boldsymbol{x}_i, \cdot), \qquad \forall n.$$

Memory and computational load grows unbounded as $n \to \infty$!

Additionally constrain the norm of f_n by a predefined $\delta > 0$:

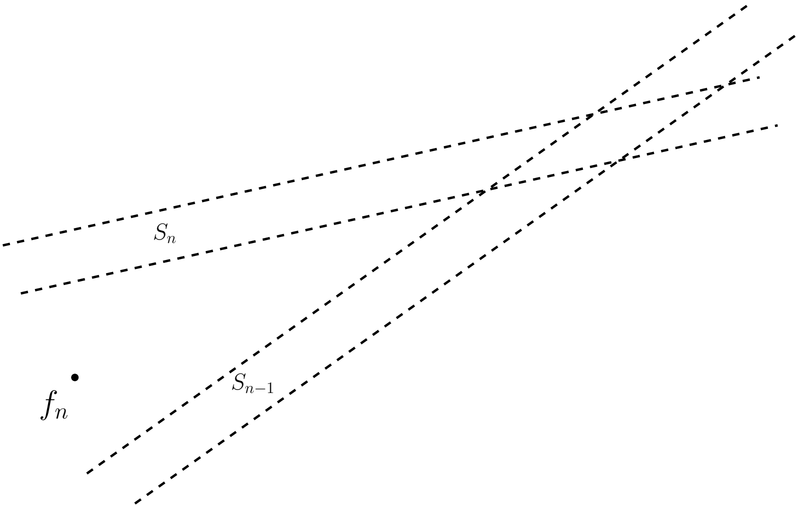
$$(\forall n \geq 0) \ f_n \in \mathcal{B} := \{f \in \mathcal{H} : \|f\| \leq \delta\} : \text{ Closed Ball.}$$

Goal

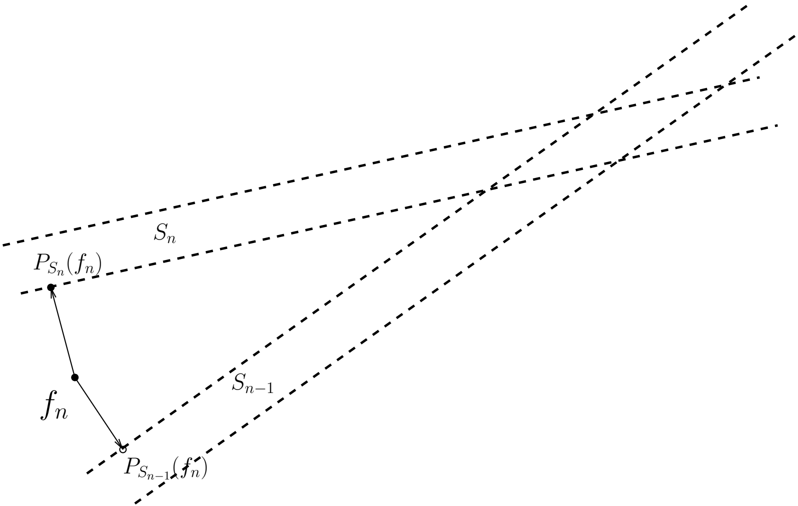
Thus, we are looking for a classifier $f \in \mathcal{H}$ such that

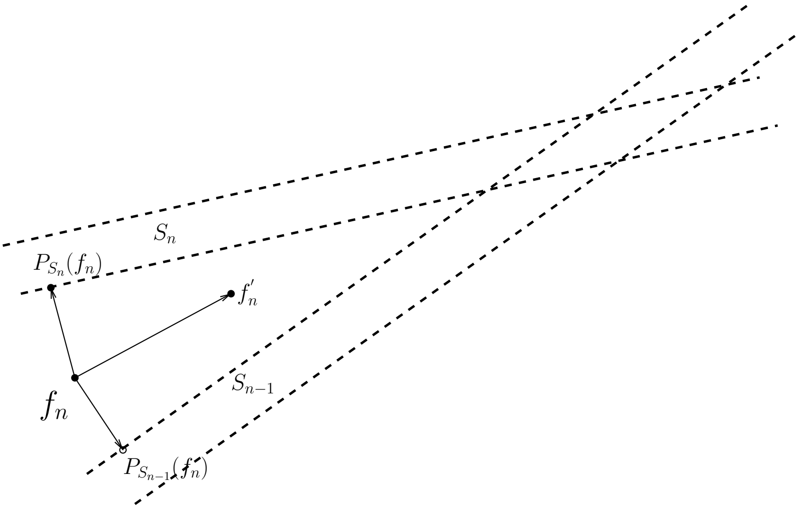
$$f \in \mathcal{B} \cap K \cap \Pi \cap (\bigcap_n S_n).$$

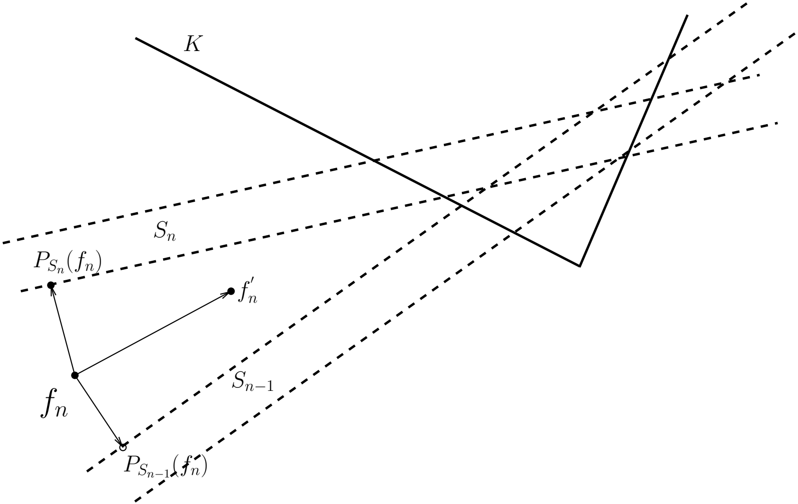
f_n

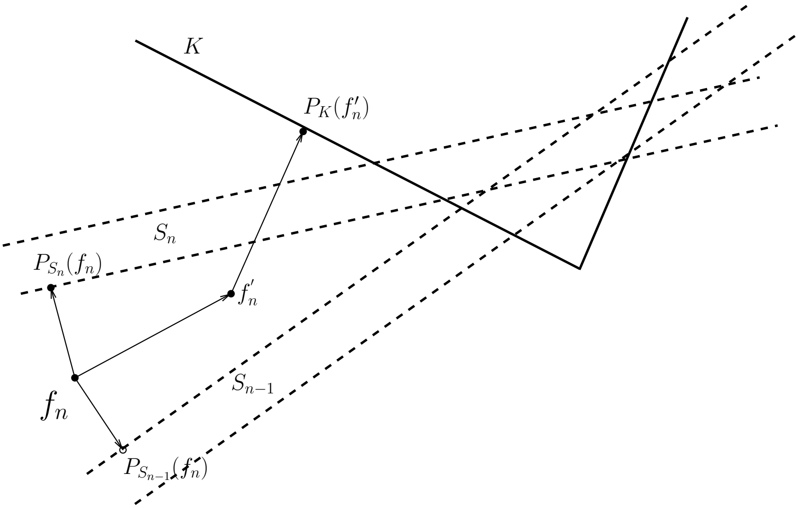


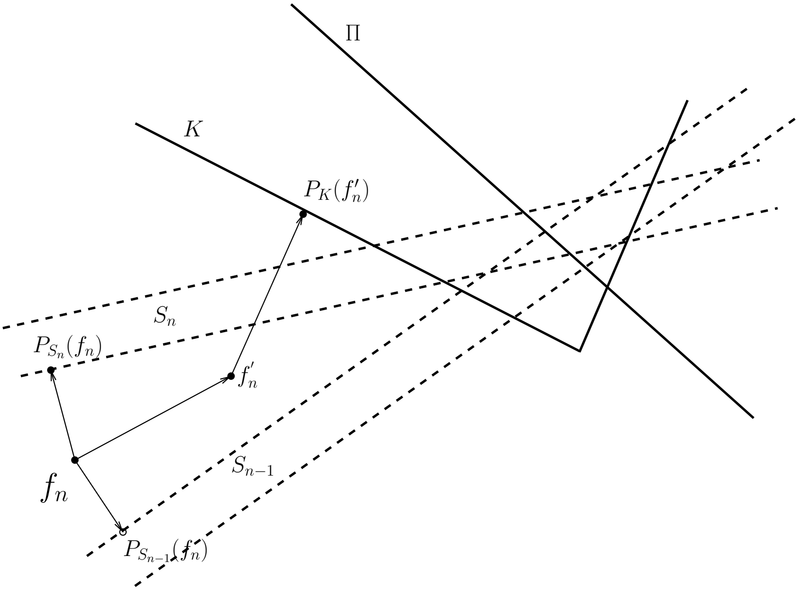
S_n
S_{n-1}
f_n

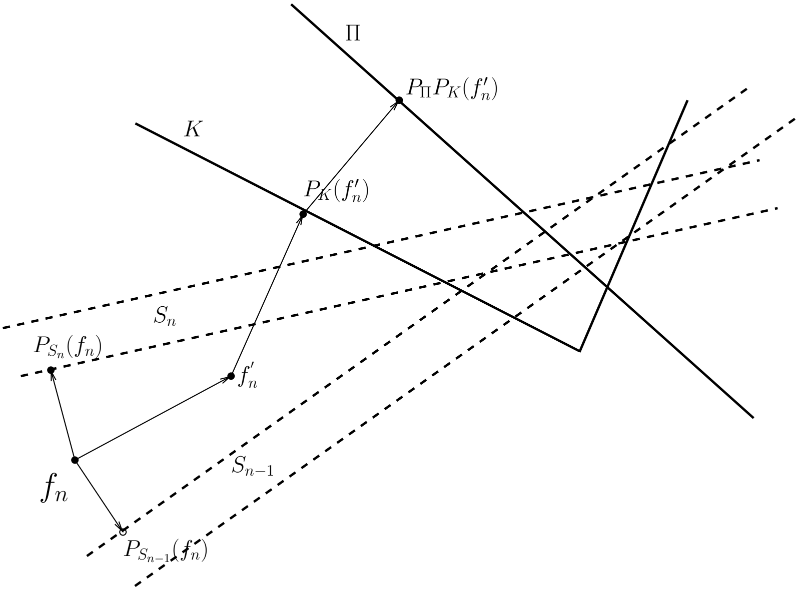

S_n
$P_{S_n}(f_n)$
S_{n-1}
f_n
$P_{S_{n-1}}(f_n)$

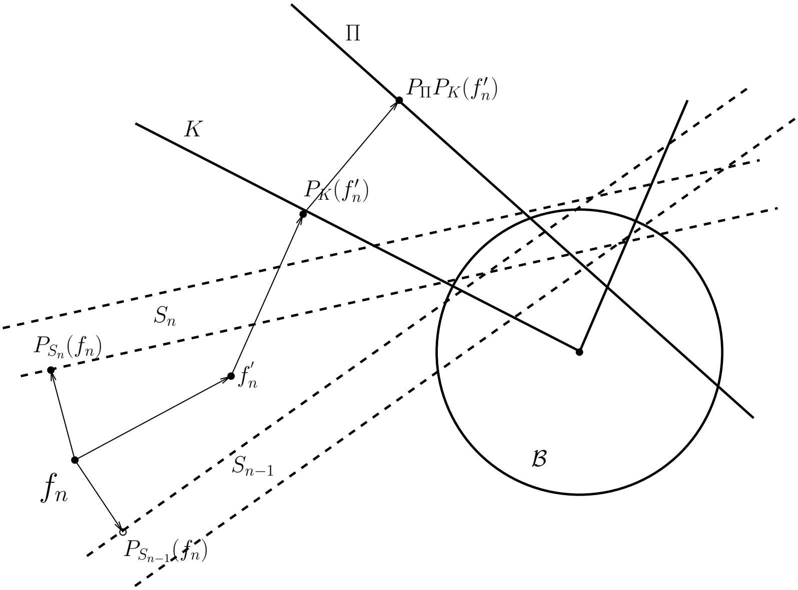

S_n
$P_{S_n}(f_n)$
f_n'
f_n
S_{n-1}
$P_{S_{n-1}}(f_n)$

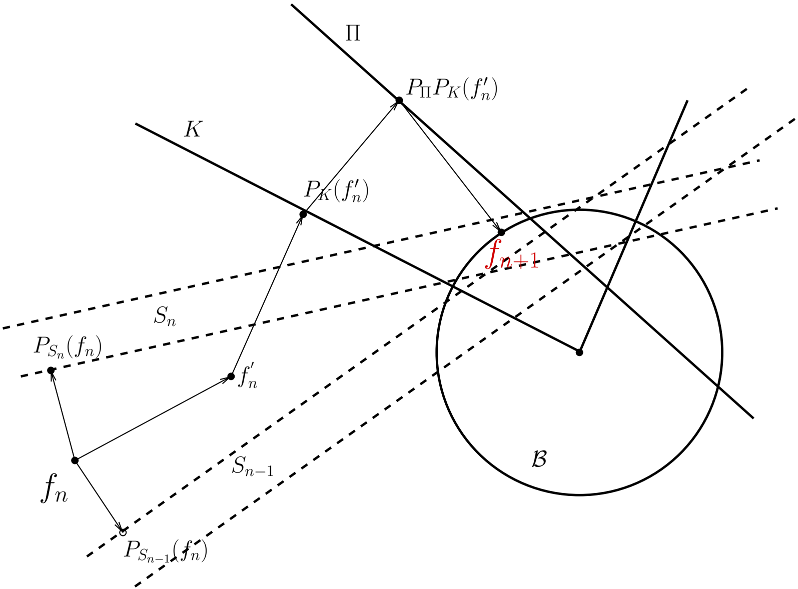

K
S_n
$P_{S_n}(f_n)$
f_n'
S_{n-1}
f_n
$P_{S_{n-1}}(f_n)$


K
$P_K(f_n')$
S_n
$P_{S_n}(f_n)$
f_n'
f_n
S_{n-1}
$P_{S_{n-1}}(f_n)$


Π
K
$P_K(f'_n)$
S_n
$P_{S_n}(f_n)$
f'_n
S_{n-1}
f_n
$P_{S_{n-1}}(f_n)$


Π
$P_\Pi P_K(f_n')$
K
$P_K(f_n')$
S_n
$P_{S_n}(f_n)$
f_n'
f_n
S_{n-1}
$P_{S_{n-1}}(f_n)$


Π
$P_{\Pi}P_K(f_n')$
K
$P_K(f_n')$
S_n
$P_{S_n}(f_n)$
f_n'
S_{n-1}
f_n
$P_{S_{n-1}}(f_n)$
$\mathcal{B}$


Π
$P_\Pi P_K(f'_n)$
K
$P_K(f'_n)$
f_{n+1}
S_n
$P_{S_n}(f_n)$
f'_n
S_{n-1}
$\mathcal{B}$
f_n
$P_{S_{n-1}}(f_n)$

The quadratic ϵ-insensitive loss function case

$$\Theta_n(f) := \max\{0, (\langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - y_n)^2 - \epsilon\}, \qquad \forall f \in \mathcal{H}, \forall n.$$

The quadratic ϵ-insensitive loss function case

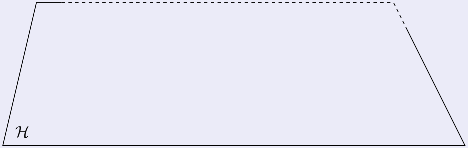
$$\Theta_n(f) := \max\{0, (\langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - y_n)^2 - \epsilon\}, \qquad \forall f \in \mathcal{H}, \forall n.$$

Piece of Information: $C_n := \{f \in \mathcal{H} : \Theta_n(f) \leq 0\}$.

The quadratic ϵ-insensitive loss function case

$$\Theta_n(f) := \max\{0, (\langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - y_n)^2 - \epsilon\}, \qquad \forall f \in \mathcal{H}, \forall n.$$

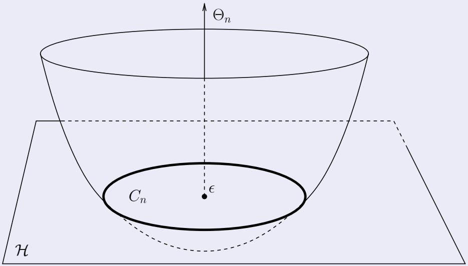
Piece of Information: $C_n := \{f \in \mathcal{H} : \Theta_n(f) \leq 0\}$.



The quadratic ϵ-insensitive loss function case

$$\Theta_n(f) := \max\{0, (\langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - y_n)^2 - \epsilon\}, \qquad \forall f \in \mathcal{H}, \forall n.$$

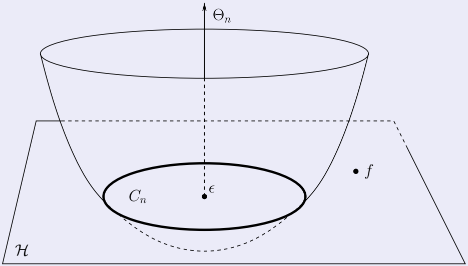
Piece of Information: $C_n := \{f \in \mathcal{H} : \Theta_n(f) \leq 0\}$.



The quadratic ϵ-insensitive loss function case

$$\Theta_n(f) := \max\{0, (\langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - y_n)^2 - \epsilon\}, \qquad \forall f \in \mathcal{H}, \forall n.$$

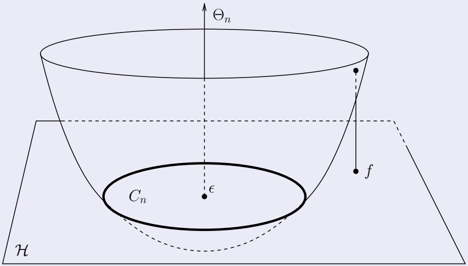
Piece of Information: $C_n := \{f \in \mathcal{H} : \Theta_n(f) \leq 0\}$.



The quadratic ϵ-insensitive loss function case

$$\Theta_n(f) := \max\{0, (\langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - y_n)^2 - \epsilon\}, \qquad \forall f \in \mathcal{H}, \forall n.$$

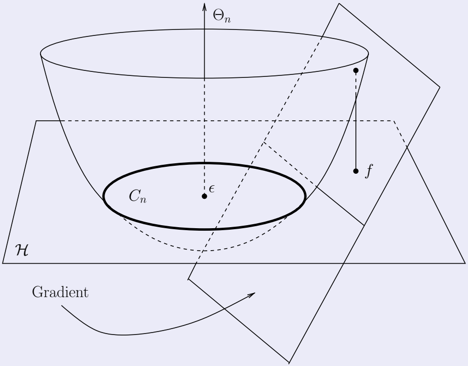
Piece of Information: $C_n := \{f \in \mathcal{H} : \Theta_n(f) \leq 0\}$.



The quadratic ϵ-insensitive loss function case

$$\Theta_n(f) := \max\{0, (\langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - y_n)^2 - \epsilon\}, \qquad \forall f \in \mathcal{H}, \forall n.$$

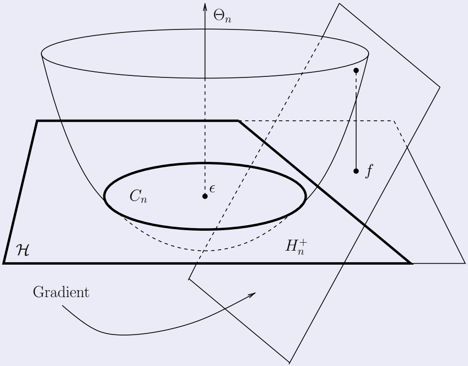
Piece of Information: $C_n := \{f \in \mathcal{H} : \Theta_n(f) \leq 0\}$.



The quadratic ϵ-insensitive loss function case

$$\Theta_n(f) := \max\{0, (\langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - y_n)^2 - \epsilon\}, \qquad \forall f \in \mathcal{H}, \forall n.$$

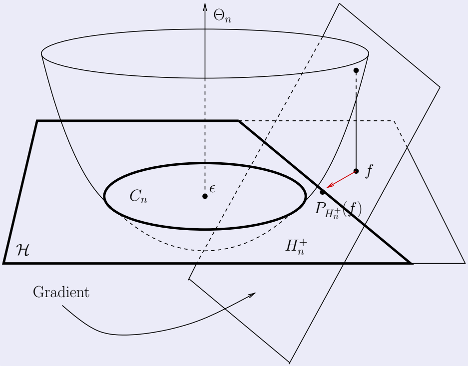
Piece of Information: $C_n := \{f \in \mathcal{H} : \Theta_n(f) \leq 0\}$.



The quadratic ϵ-insensitive loss function case

$$\Theta_n(f) := \max\{0, (\langle f, \kappa(\boldsymbol{x}_n, \cdot)\rangle - y_n)^2 - \epsilon\}, \qquad \forall f \in \mathcal{H}, \forall n.$$

Piece of Information: $C_n := \{f \in \mathcal{H} : \Theta_n(f) \leq 0\}$.



$$P_{H_n^+}(f) = f - \lambda_n \frac{\Theta_n(f)}{\|\Theta_n'(f)\|^2}\Theta_n'(f).$$

The Recursion

For an arbitrary $f_0 \in \mathcal{H}$, and $\forall n$,

$$f_{n+1} = \begin{cases} T\left(f_n - \lambda_n \frac{\Theta_n(f_n)}{\|\Theta_n'(f_n)\|^2}\Theta_n'(f_n)\right), & \text{if } \Theta_n'(f_n) \neq 0, \\ T(f_n), & \text{if } \Theta_n'(f_n) = 0, \end{cases}$$

where

The Recursion

For an arbitrary $f_0 \in \mathcal{H}$, and $\forall n$,

$$f_{n+1} = \begin{cases} T\left(f_n - \lambda_n \frac{\Theta_n(f_n)}{\|\Theta_n'(f_n)\|^2} \Theta_n'(f_n)\right), & \text{if } \Theta_n'(f_n) \neq 0, \\ T(f_n), & \text{if } \Theta_n'(f_n) = 0, \end{cases}$$

where

- T comprises the projections associated with the constraints.

The Recursion

For an arbitrary $f_0 \in \mathcal{H}$, and $\forall n$,

$$f_{n+1} = \begin{cases} T\left(f_n - \lambda_n \frac{\Theta_n(f_n)}{\|\Theta_n'(f_n)\|^2}\Theta_n'(f_n)\right), & \text{if } \Theta_n'(f_n) \neq 0, \\ T(f_n), & \text{if } \Theta_n'(f_n) = 0, \end{cases}$$

where

- T comprises the projections associated with the constraints.
- In case Θ_n is non-differentiable the subgradient Θ_n' is used in the place of the gradient.

The Recursion

For an arbitrary $f_0 \in \mathcal{H}$, and $\forall n$,

$$f_{n+1} = \begin{cases} T\left(f_n - \lambda_n \frac{\Theta_n(f_n)}{\|\Theta'_n(f_n)\|^2}\Theta'_n(f_n)\right), & \text{if } \Theta'_n(f_n) \neq 0, \\ T(f_n), & \text{if } \Theta'_n(f_n) = 0, \end{cases}$$

where

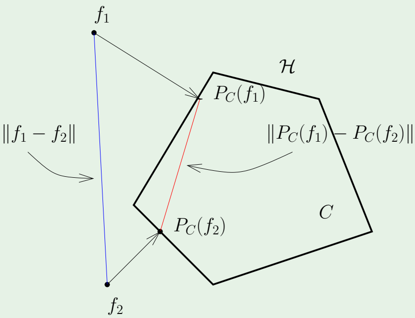
- T comprises the projections associated with the constraints.
- In case Θ_n is non-differentiable the subgradient Θ'_n is used in the place of the gradient.
- Note that the above recursion holds true for any strongly attracting nonexpansive mapping T [Slavakis, Yamada, Ogura '06].

Definition (Nonexpansive Mapping)

A mapping T is called nonexpansive if

$$\|T(f_1) - T(f_2)\| \leq \|f_1 - f_2\|, \qquad \forall f_1, f_2 \in \mathcal{H}.$$

Example (Projection Mapping)

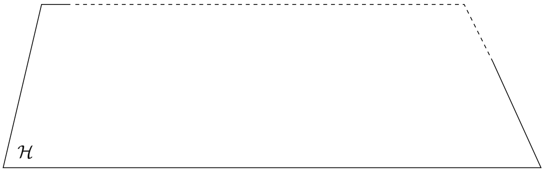


Nondifferentiable Loss Function

Definition (Subgradient)

Given a convex continuous function Θ_n, the subgradient $\Theta_n'(f)$ is an element of $\mathcal{H}$ such that

$$\langle g - f, \Theta_n'(f) \rangle + \Theta_n(f) \leq \Theta_n(g), \forall g \in \mathcal{H}.$$

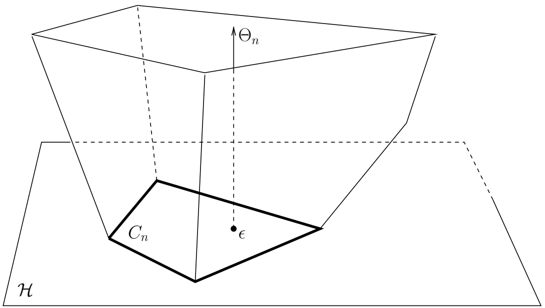


Nondifferentiable Loss Function

Definition (Subgradient)

Given a convex continuous function Θ_n, the subgradient $\Theta_n'(f)$ is an element of $\mathcal{H}$ such that

$$\langle g - f, \Theta_n'(f)\rangle + \Theta_n(f) \leq \Theta_n(g), \forall g \in \mathcal{H}.$$

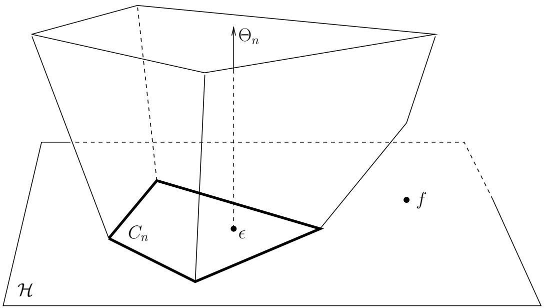


Nondifferentiable Loss Function

Definition (Subgradient)

Given a convex continuous function Θ_n, the subgradient $\Theta'_n(f)$ is an element of $\mathcal{H}$ such that

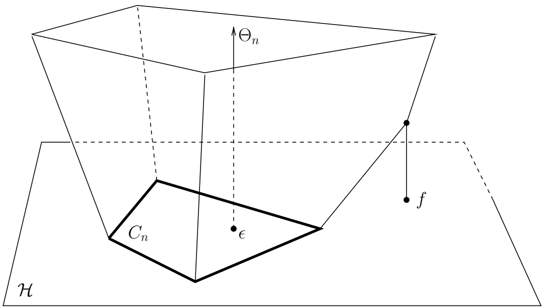
$$\langle g - f, \Theta'_n(f)\rangle + \Theta_n(f) \leq \Theta_n(g), \forall g \in \mathcal{H}.$$



Definition (Subgradient)

Given a convex continuous function Θ_n, the subgradient $\Theta_n'(f)$ is an element of $\mathcal{H}$ such that

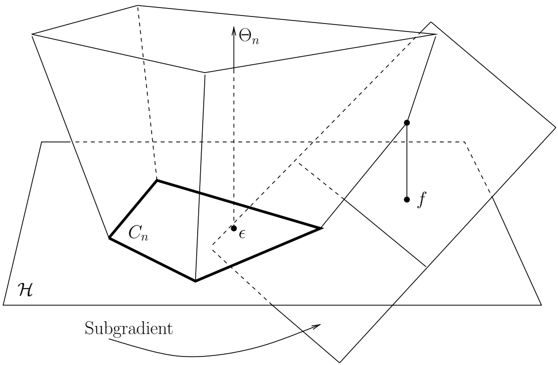
$$\langle g - f, \Theta_n'(f)\rangle + \Theta_n(f) \leq \Theta_n(g), \forall g \in \mathcal{H}.$$



Definition (Subgradient)

Given a convex continuous function Θ_n, the subgradient $\Theta'_n(f)$ is an element of $\mathcal{H}$ such that

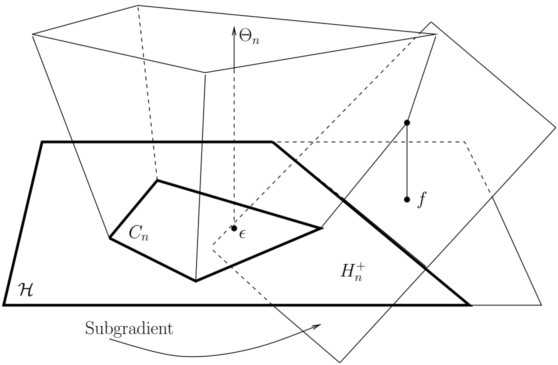
$$\langle g - f, \Theta'_n(f) \rangle + \Theta_n(f) \leq \Theta_n(g), \forall g \in \mathcal{H}.$$



Definition (Subgradient)

Given a convex continuous function Θ_n, the subgradient $\Theta_n'(f)$ is an element of $\mathcal{H}$ such that

$$\langle g - f, \Theta_n'(f) \rangle + \Theta_n(f) \le \Theta_n(g), \forall g \in \mathcal{H}.$$

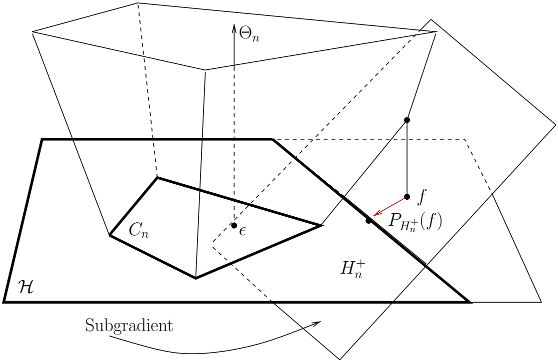


Nondifferentiable Loss Function

Definition (Subgradient)

Given a convex continuous function Θ_n, the subgradient $\Theta'_n(f)$ is an element of $\mathcal{H}$ such that

$$\langle g - f, \Theta'_n(f)\rangle + \Theta_n(f) \le \Theta_n(g), \forall g \in \mathcal{H}.$$



Definition (Fixed Point Set)

Given a mapping $T : \mathcal{H} \to \mathcal{H}$, $\mathrm{Fix}(T) := \{f \in \mathcal{H} : T(f) = f\}$.

Define at $n \geq 0$, $\Omega_n := \mathrm{Fix}(T) \cap (\arg\min_{f \in \mathcal{H}} \Theta_n(f))$. Let $\Omega := \bigcap_{n \geq n_0} \Omega_n \neq \emptyset$, for some nonnegative integer n_0. Set the extrapolation parameter $\mu_n \in [\mathcal{M}_n \epsilon_1, \mathcal{M}_n(2 - \epsilon_2)]$, $\forall n \geq n_0$ for some sufficiently small $\epsilon_1, \epsilon_2 > 0$. Then, the following statements hold.

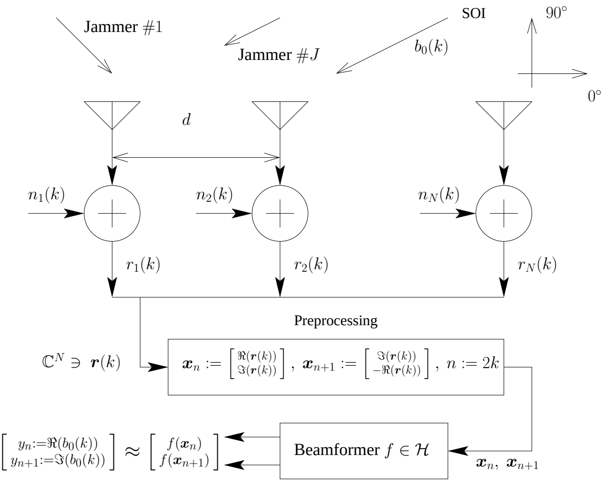
- Monotone approximation. For any $f' \in \Omega$, we have

$$\|f_{n+1} - f'\| \leq \|f_n - f'\|, \quad \forall n \geq n_0.$$

- Asymptotic minimization. $\lim_{n \to \infty} \Theta_n(f_n) = 0$.
- Strong convergence. Assume that there exists a hyperplane $\Pi \subset \mathcal{H}$ such that $\mathrm{ri}_{\Pi}(\Omega) \neq \emptyset$. Then, there exists a $f_* \in \mathrm{Fix}(T)$ such that $\lim_{n \to \infty} f_n =: f_*$.
- Characterization of the limit point. Assume that $\mathrm{int}(\Omega) \neq \emptyset$. Then, the limit point

$$f_* \in \mathrm{clos}(\liminf_{n \to \infty} \Omega_n),$$

where $\liminf_{n \to \infty} \Omega_n := \bigcup_{m=0}^{\infty} \bigcap_{n \geq m} \Omega_n$.



$$\boldsymbol{r}(k) := \sum_{l=0}^{J} \alpha_l b_l(k) \boldsymbol{s}_l + \boldsymbol{n}(k), \ \forall k \geq 0,$$

$\boldsymbol{s}_l$: Steering vectors.

Problem Formulation

- Training Data: The received signals and the sequence of symbols sent by the Signal Of Interest (SOI).

- Training Data: The received signals and the sequence of symbols sent by the Signal Of Interest (SOI).
- Constraints: Given erroneous information $\tilde{\boldsymbol{s}}_0$ on the actual SOI steering vector $\boldsymbol{s}_0$ (e.g. imperfect array calibration), find a solution that gives uniform output for all the steering vectors in an area around $\tilde{\boldsymbol{s}}_0$; use a closed ball $B[\tilde{\boldsymbol{s}}_0, \delta]$.

$$\Downarrow$$

Robustness is desired!

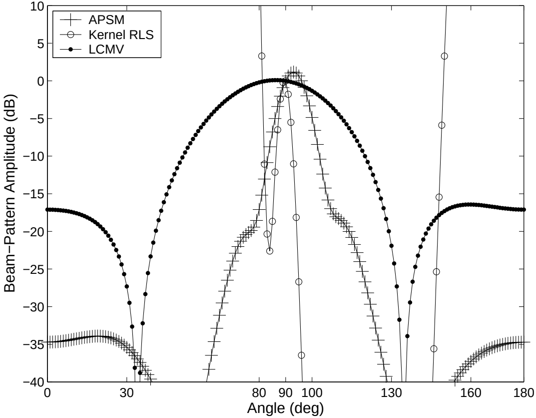
Problem Formulation

- Training Data: The received signals and the sequence of symbols sent by the Signal Of Interest (SOI).
- Constraints: Given erroneous information $\tilde{\boldsymbol{s}}_0$ on the actual SOI steering vector $\boldsymbol{s}_0$ (e.g. imperfect array calibration), find a solution that gives uniform output for all the steering vectors in an area around $\tilde{\boldsymbol{s}}_0$; use a closed ball $B[\tilde{\boldsymbol{s}}_0, \delta]$.

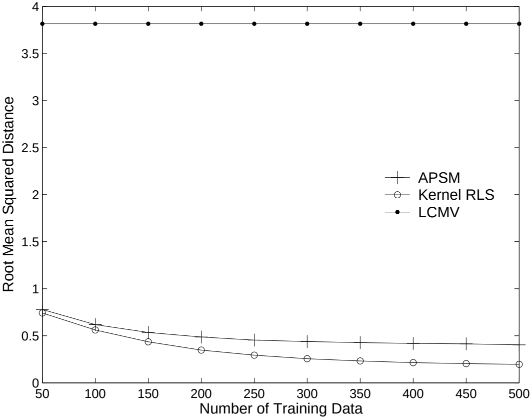
$$\Downarrow$$

Robustness is desired!

- Antenna Geometry: Only 3 array elements, but with 5 jammers with SNRs $10, 30, 20, 10$, and 30 dB. The SOI's SNR is set equal to 10 dB.



	Input	LCMV	KRLS	APSM
SINR (dB)	-23.26	-20.21	Very low	18.65


4
3.5
3
2.5
2
1.5
1
0.5
0
Root Mean Squared Distance
50
100
150
200
250
300
350
400
450
500
Number of Training Data
APSM
Kernel RLS
LCMV

Conclusions

- A geometric framework for learning in Reproducing Kernel Hilbert Spaces (RKHS) was presented.
- The key ingredients of the framework are
  - the basic tool of metric projections,
  - the Set Theoretic Estimation approach, where each property of the system is described by a closed convex set.
- Both the online classification and regression tasks were considered.
- The way to encapsulate a-priori constraints as well as sparsification, in the framework was also depicted.
- The framework can be easily extended to any continuous, not necessarily differentiable, convex cost function, and to any closed convex a-priori constraint.
- A nonlinear online beamforming task was presented in order to validate the proposed approach.