Adaptive Processing in a World of Projections

Sergios Theodoridis

Joint work with
Konstantinos Slavakis and Isao Yamada

1University of Athens, Greece
2University of Peloponnese, Greece
3Tokyo Institute of Technology, Japan

January 16, 2009
“ὍΔΕΙΣ ΑΓΕΩΜΕΤΡΗΤΟΣ ΕΙΣΙ”
“ΟΥΔΕΙΣ ΑΓΕΩΜΕΤΡΗΤΟΣ ΕΙΣΙ”

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

Plato’s Academy of Philosophy
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 \((x_n, y_n)_{n=1}^{N}\), which are jointly distributed, and
- A parametric set of functions

\[ F = \{ f_\alpha(x) : \alpha \in A \subset \mathbb{R}^k \} . \]

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

\[ y \approx f(x) . \]
Problem Definition

Given

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

\[
\mathcal{F} = \{f_\alpha(x) : \alpha \in A \subseteq \mathbb{R}^k\}.
\]

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

\[
y \approx f(x).
\]

Special Cases

Smoothing, prediction, filtering, system identification, beamforming, curve-fitting, regression, and classification.
Select a loss function $\ell(\cdot, \cdot)$ and estimate $f(\cdot)$ so that

$$f(\cdot) \in \{ f_\alpha(\cdot) \in \arg\min_\alpha \sum_{n=1}^{N} \ell(y_n, f_\alpha(x_n)) \}.$$
The More Classical Approach

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

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

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 \textit{closest to} $f$ (Pythagoras Theorem).
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

$$\|f - f^*\| = \min_{g \in C} \|f - g\|.$$
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 $P_C : \mathcal{H} \to C : f \mapsto f_*$. 

![Diagram of projection onto a closed convex set](image.png)
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 $P_C : \mathcal{H} \to C : f \mapsto f_*$. 

![Diagram of projection onto a closed convex set](image-url)
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 $H := \{ g \in \mathcal{H} : \langle g, a \rangle = c \}$)

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

$P_H(f) = f - \frac{\langle f, a \rangle - c}{\|a\|^2} a,$ \quad \forall f \in \mathcal{H}.$
Example (Halfspace $H^{-} := \{ g \in \mathcal{H} : \langle g, a \rangle \leq c \}$)
Example (Halfspace $H^- := \{ g \in \mathcal{H} : \langle g, a \rangle \leq c \}$)

$$P_{H^-}(f) = f - \frac{\max\{0, \langle f, a \rangle - c\}}{\|a\|^2}a, \quad \forall f \in \mathcal{H}.$$
Example (Closed Ball $B[0, \delta] := \{g \in \mathcal{H} : \|g\| \leq \delta\}$)
Example (Closed Ball) \[ B[0, \delta] := \{ g \in \mathcal{H} : \|g\| \leq \delta \} \]

\[ P_{B[0,\delta]}(f) := \begin{cases} f, & \text{if } \|f\| \leq \delta, \\ \frac{\delta}{\|f\|} f, & \text{if } \|f\| > \delta. \end{cases} \quad \forall f \in \mathcal{H}. \]
Example (Icecream Cone $K := \{(f, \tau) \in \mathcal{H} \times \mathbb{R} : \|f\| \geq \tau\}$)
Example (Icecream Cone $K := \{(f, \tau) \in \mathcal{H} \times \mathbb{R} : \|f\| \geq \tau\}$)

$$P_K((f, \tau)) = \begin{cases} (f, \tau), & \text{if } \|f\| \leq \tau, \\ (0, 0), & \text{if } \|f\| \leq -\tau, \\ \frac{\|f\| + \tau}{2} \left(\frac{f}{\|f\|}, 1\right), & \text{otherwise,} \end{cases}$$

$\forall (f, \tau) \in \mathcal{H} \times \mathbb{R}.$
Relaxed Projection

**Definition**

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

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

**Definition**

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

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

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

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

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

Remark: The use of the relaxed projection operator with $\mu > 1$ can, potentially, speed up the convergence rate of the algorithms to be presented.
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:
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 $f \in \mathcal{H}$, $\lim_{n \to \infty} (P_{M_2} P_{M_1})^n(f) = P_{M_1 \cap M_2}(f)$. 
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

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

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

$$f_{n+1} := T_{C_q} \cdots T_{C_1}(f_n), \quad \forall n.$$
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 $f_0 \in H$, this defines the sequence of points

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

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

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

\[
f_{n+1} := T_{C_q} \cdots T_{C_1}(f_n), \quad \forall n.
\]
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

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

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

\[
f_{n+1} := T_{C_q} \cdots T_{C_1}(f_n), \quad \forall n.
\]

\( C_1 \cap C_2 \)
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

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

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

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

**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

\[
 f_{n+1} = f_n + \mu_n \left( \sum_{i=1}^{q} w_i P_{C_i}(f_n) - f_n \right), \quad \forall n,
\]

Convex combination of projections
Recall

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

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

\[ f_{n+1} = f_n + \mu_n \left( \sum_{i=1}^{q} w_i P_{C_i}(f_n) - f_n \right), \forall n, \]

Convex combination of projections
Recall

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

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

\[ f_{n+1} = f_n + \mu_n \left( \sum_{i=1}^{q} w_i P_{C_i}(f_n) - f_n \right), \forall n, \]

Convex combination of projections
Recall

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

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

\[ f_{n+1} = f_n + \mu_n \left( \sum_{i=1}^q w_i P_{C_i}(f_n) - f_n \right), \quad \forall n, \]

Convex combination of projections
Recall

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

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

\[ f_{n+1} = f_n + \mu_n \left( \sum_{i=1}^{q} w_i P_{C_i}(f_n) - f_n \right), \forall n, \]

Convex combination of projections
Recall

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

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

\[ f_{n+1} = f_n + \mu_n \left( \sum_{i=1}^{q} w_i P_{C_i}(f_n) - f_n \right), \forall n, \]

converges weakly to a point \( f_* \) in \( \bigcap_{i=1}^{q} C_i \), where \( \mu_n \in (\epsilon, M_n) \), for \( \epsilon \in (0, 1) \), and

\[ M_n := \frac{\sum_{i=1}^{q} w_i \| P_{C_i}(f_n) - f_n \|^2}{\| \sum_{i=1}^{q} w_i P_{C_i}(f_n) - f_n \|^2}. \]
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

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

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

where \(\mu_n\) is the step size, \(w_j\) are weights, and \(P_{C_j}\) are projection mappings.
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

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

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

where \(\mu_n\) is the step size.
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

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

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

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

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

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

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

Given a set of training samples \( x_0, \ldots, x_N \subset \mathbb{R}^m \) and a set of corresponding desired responses \( y_0, \ldots, y_N \), estimate a function \( f(\cdot) : \mathbb{R}^m \rightarrow \mathbb{R} \) that fits the data.
The Task

Given a set of training samples \( x_0, \ldots, x_N \subset \mathbb{R}^m \) 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:

\[
\min_f \mathbb{E}\{\ell(f(x), y)\},
\]

or, in practice, the empirical risk is minimized:

\[
\min_f \sum_{n=0}^N \ell(f(x_n), y_n).
\]
Loss Functions

Example (Classification)

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

$$\ell(f(x_n), y_n) := \max\{0, \rho - y_n f(x_n)\}, \quad \forall n.$$
Example (Regression)

The square loss functions:

\[ \ell(f(x_n), y_n) := (y_n - f(x_n))^2, \quad \forall n. \]
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 \((x_n, y_n)\), 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 \((x_n, y_n)\), 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 \((x_n, y_n)\), 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.
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 \((x_n, y_n)\), 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

$$f \in \bigcap_{n} C_n \subset \mathcal{H}.$$
The Setting

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

\[
\begin{align*}
  y_n &= +1, \quad x_n \in W_1, \\
  y_n &= -1, \quad x_n \in W_2.
\end{align*}
\]

Assume linear separable classes.
The Setting

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

\[
\begin{cases}
y_n = +1, & x_n \in W_1, \\
y_n = -1, & x_n \in W_2.
\end{cases}
\]

Assume linear separable classes.

The Goal (for \(\rho = 0\))
The Setting

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

\[
\begin{align*}
y_n &= +1, \quad x_n \in W_1, \\
y_n &= -1, \quad x_n \in W_2.
\end{align*}
\]

Assume linear separable classes.

The Goal (for \(\rho = 0\))

Find \(f(x) = \mathbf{w}^t x + b\), so that

\[
\begin{align*}
\mathbf{w}^t x_n + b &\geq 0, \quad \text{if } y_n = +1, \\
\mathbf{w}^t x_n + b &\leq 0, \quad \text{if } y_n = -1.
\end{align*}
\]
The Setting

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

Assume the two class task,

\[
\begin{cases}
  y_n = +1, & x_n \in W_1, \\
  y_n = -1, & x_n \in W_2.
\end{cases}
\]

Assume linear separable classes.

The Goal (for \(\rho = 0\))

Find \(f(x) = w^t x + b\), so that

\[
\begin{cases}
  w^t x_n + b \geq 0, & \text{if } y_n = +1, \\
  w^t x_n + b \leq 0, & \text{if } y_n = -1.
\end{cases}
\]

Hereafter, \((w \leftarrow [w]_b, x_n \leftarrow [x^n]_1)\).
Find all those $w$ so that $y_n w^t x_n \geq 0$, $n = 0, 1, \ldots$
Find all those $w$ so that $y_n w^t x_n \geq 0$, $n = 0, 1, \ldots$

The Equivalent Set

$$H^+_n := \{ w \in \mathbb{R}^m : y_n x_n^t w \geq 0 \}, \ n = 0, 1, \ldots$$
The feasibility set

For each pair \((x_n, y_n)\), form the equivalent halfspace \(H_n^+\), and find \(w* \in \bigcap_n H_n^+\).

If linearly separable, the problem is feasible.
The feasibility set

For each pair \((x_n, y_n)\), form the equivalent halfspace \(H_n^+\), and

\[
\text{find } w_* \in \bigcap_n H_n^+. 
\]

If linearly separable, the problem is feasible.

The Algorithm

Each \(H_n^+\) is a convex set.

- Start from an arbitrary initial \(w_0\).
The feasibility set
For each pair \((x_n, y_n)\), form the equivalent halfspace \(H_n^+\), and
\[ \text{find } w_* \in \bigcap_n H_n^+ . \]

If linearly separable, the problem is feasible.

The Algorithm

Each \(H_n^+\) is a convex set.

- Start from an arbitrary initial \(w_0\).
- Keep projecting as each \(H_n^+\) is formed.
The feasibility set

For each pair \((x_n, y_n)\), form the equivalent halfspace \(H_n^+\), and

\[ \text{find } w_* \in \bigcap_n H_n^+. \]

If linearly separable, the problem is feasible.

The Algorithm

Each \(H_n^+\) is a convex set.

- Start from an arbitrary initial \(w_0\).
- Keep projecting as each \(H_n^+\) is formed.

\[ P_{H_n^+}(w) = w - \frac{\min\{0, \langle w, y_n x_n \rangle\}}{\|x_n\|^2} y_n x_n, \quad \forall w \in \mathcal{H}. \]
The feasibility set

For each pair \((x_n, y_n)\), form the equivalent halfspace \(H_n^+\), and

\[
\text{find } w_* \in \bigcap_{n} H_n^+.
\]

If linearly separable, the problem is feasible.

The Algorithm

Each \(H_n^+\) is a convex set.

- Start from an arbitrary initial \(w_0\).
- Keep projecting as each \(H_n^+\) is formed.

\[
P_{H_n^+}(w) = w - \frac{\min\{0, \langle w, y_n x_n \rangle\}}{\|x_n\|^2} y_n x_n,
\]

\(\forall w \in \mathcal{H}\).
The feasibility set

For each pair \((x_n, y_n)\), form the equivalent halfspace \(H_n^+\), and find \(w_* \in \bigcap_n H_n^+\).

If linearly separable, the problem is feasible.

The Algorithm

Each \(H_n^+\) is a convex set.

- Start from an arbitrary initial \(w_0\).
- Keep projecting as each \(H_n^+\) is formed.
- \(P_{H_n^+}(w) = w - \frac{\min\{0, \langle w, y_n x_n \rangle\}}{\|x_n\|^2} y_n x_n\), \(\forall w \in \mathcal{H}\).
The feasibility set

For each pair \((x_n, y_n)\), form the equivalent halfspace \(H^+_n\), and

\[
\text{find } w_* \in \bigcap_n H^+_n.
\]

If linearly separable, the problem is feasible.

The Algorithm

Each \(H^+_n\) is a convex set.

- Start from an arbitrary initial \(w_0\).
- Keep projecting as each \(H^+_n\) is formed.

\[
P_{H^+_n}(w) = w - \frac{\min\{0, \langle w, y_n x_n \rangle\}}{\|x_n\|^2} y_n x_n, \quad \forall w \in \mathcal{H}.
\]
The feasibility set

For each pair \((x_n, y_n)\), form the equivalent halfspace \(H_n^+\), and

\[
\text{find } w_\ast \in \bigcap_n H_n^+.
\]

If linearly separable, the problem is feasible.

The Algorithm

Each \(H_n^+\) is a convex set.

- Start from an arbitrary initial \(w_0\).
- Keep projecting as each \(H_n^+\) is formed.

\[
P_{H_n^+}(w) = w - \frac{\min\{0, \langle w, y_n x_n \rangle\}}{\|x_n\|^2} y_n x_n, \quad \forall w \in \mathcal{H}.
\]
The feasibility set

For each pair \((x_n, y_n)\), form the equivalent halfspace \(H_n^+\), and

\[
\text{find } w_* \in \bigcap_{n} H_n^+.
\]

If linearly separable, the problem is feasible.

The Algorithm

Each \(H_n^+\) is a convex set.

- Start from an arbitrary initial \(w_0\).
- Keep projecting as each \(H_n^+\) is formed.

\[
P_{H_n^+}(w) = w - \frac{\min\{0, \langle w, y_n x_n \rangle\}}{\|x_n\|^2} y_n x_n, \quad \forall w \in \mathcal{H}.
\]
The feasibility set

For each pair \((x_n, y_n)\), form the equivalent halfspace \(H_n^+\), and

\[
\text{find } w_* \in \bigcap_{n} H_n^+.
\]

If linearly separable, the problem is feasible.

The Algorithm

Each \(H_n^+\) is a convex set.
- Start from an arbitrary initial \(w_0\).
- Keep projecting as each \(H_n^+\) is formed.

\[
P_{H_n^+}(w) = w - \frac{\min\{0, \langle w, y_n x_n \rangle\}}{\|x_n\|^2} y_n x_n, \quad \forall w \in \mathcal{H}.
\]
The feasibility set

For each pair \((x_n, y_n)\), form the equivalent halfspace \(H_n^+\), and

\[
\text{find } w_* \in \bigcap_n H_n^+.
\]

If linearly separable, the problem is feasible.

The Algorithm

Each \(H_n^+\) is a convex set.

- Start from an arbitrary initial \(w_0\).
- Keep projecting as each \(H_n^+\) is formed.

\[
P_{H_n^+}(w) = w - \frac{\min\{0, \langle w, y_n x_n \rangle\}}{\|x_n\|^2} y_n x_n, \quad \forall w \in \mathcal{H}.
\]
Algorithmic Solution to Online Classification

\[ \mathbf{w}_{n+1} := \mathbf{w}_n + \mu_n \left( \sum_{j \in \{n-q+1,\ldots,n\}} \omega_j^{(n)} P_{H_n^+}(\mathbf{w}_n) - \mathbf{w}_n \right), \]

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

\[ \mathcal{M}_n := \begin{cases} \frac{\sum_{j \in \{n-q+1,\ldots,n\}} \omega_j^{(n)} \| P_{H_n^+}(\mathbf{w}_n) - \mathbf{w}_n \|^2}{\| \sum_{j \in \{n-q+1,\ldots,n\}} \omega_j^{(n)} P_{H_n^+}(\mathbf{w}_n) - \mathbf{w}_n \|^2}, & \text{if } \mathbf{w}_n \notin \bigcap_{j \in \{n-q+1,\ldots,n\}} H_n^+ \bigcup_n, \\ 1, & \text{otherwise}. \end{cases} \]
\[ w_{n+1} := w_n + \mu_n \left( \sum_{j \in \{n-q+1, \ldots, n\}} \omega_j^{(n)} P_{H_n^+}(w_n) - w_n \right), \]

\[ \mu_n \in [0, 2M_n], \text{ and} \]

\[ M_n := \begin{cases} \frac{\sum_{j \in \{n-q+1, \ldots, n\}} \omega_j^{(n)} \| P_{H_n^+}(w_n) - w_n \|^2}{\| \sum_{j \in \{n-q+1, \ldots, n\}} \omega_j^{(n)} P_{H_n^+}(w_n) - w_n \|^2}, & \text{if } w_n \notin \bigcap_{j \in \{n-q+1, \ldots, n\}} H_n^+, \\ 1, & \text{otherwise}. \end{cases} \]

\[ w_n. \]
Algorithmic Solution to Online Classification

\[ w_{n+1} := w_n + \mu_n \left( \sum_{j \in \{n-q+1, \ldots, n\}} \omega_j^{(n)} P_{H_n^+}(w_n) - w_n \right), \]

\[ \mu_n \in [0, 2M_n], \text{ and} \]

\[ M_n := \begin{cases} 
\frac{\sum_{j \in \{n-q+1, \ldots, n\}} \omega_j^{(n)} \| P_{H_n^+}(w_n) - w_n \|^2}{\| \sum_{j \in \{n-q+1, \ldots, n\}} \omega_j^{(n)} P_{H_n^+}(w_n) - w_n \|^2}, & \text{if } w_n \notin \bigcap_{j \in \{n-q+1, \ldots, n\}} H_n^+, \\
1, & \text{otherwise.} \end{cases} \]
\( w_{n+1} := w_n + \mu_n \left( \sum_{j \in \{n-q+1, \ldots, n\}} \omega_j^{(n)} P_{H_n^+}(w_n) - w_n \right), \)

\( \mu_n \in [0, 2M_n], \) and

\[ M_n := \begin{cases} 
\frac{\sum_{j \in \{n-q+1, \ldots, n\}} \omega_j^{(n)} \| P_{H_n^+}(w_n) - w_n \|^2}{\| \sum_{j \in \{n-q+1, \ldots, n\}} \omega_j^{(n)} P_{H_n^+}(w_n) - w_n \|^2}, & \text{if } w_n \notin \bigcap_{j \in \{n-q+1, \ldots, n\}} H_n^+; \\
1, & \text{otherwise}. 
\]
$\mathbf{w}_{n+1} := \mathbf{w}_n + \mu_n \left( \sum_{j \in \{n-q+1, \ldots, n\}} \omega_j^{(n)} P_{H_n^+}(\mathbf{w}_n) - \mathbf{w}_n \right)$,

$\mu_n \in [0, 2M_n]$, and

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

where \( \mu_n \in [0, 2M_n] \), and

\[ M_n := \begin{cases} \\
\frac{\sum_{j \in \{n-q+1, \ldots, n\}} \omega_j^{(n)} \| P_{H_n^+}(\mathbf{w}_n) - \mathbf{w}_n \|^2}{\| \sum_{j \in \{n-q+1, \ldots, n\}} \omega_j^{(n)} P_{H_n^+}(\mathbf{w}_n) - \mathbf{w}_n \|^2}, & \text{if } \mathbf{w}_n \notin \bigcap_{j \in \{n-q+1, \ldots, n\}} H_n^+ \\
1, & \text{otherwise.} 
\]
Algorithmic Solution to Online Classification

\[ \mathbf{w}_{n+1} := \mathbf{w}_n + \mu_n \left( \sum_{j \in \{n-q+1, \ldots, n\}} \omega_{j}^{(n)} P_{H_n^+}(\mathbf{w}_n) - \mathbf{w}_n \right), \]

\[ \mu_n \in [0, 2M_n] \text{, and} \]

\[ M_n := \begin{cases} 
\frac{\sum_{j \in \{n-q+1, \ldots, n\}} \omega_{j}^{(n)} \| P_{H_n^+}(\mathbf{w}_n) - \mathbf{w}_n \|^2}{\| \sum_{j \in \{n-q+1, \ldots, n\}} \omega_{j}^{(n)} P_{H_n^+}(\mathbf{w}_n) - \mathbf{w}_n \|^2}, & \text{if } \mathbf{w}_n \not\in \bigcap_{j \in \{n-q+1, \ldots, n\}} H_n^+ \text{,} \\
1, & \text{otherwise.} 
\end{cases} \]
\[ w_{n+1} := w_n + \mu_n \left( \sum_{j \in \{n-q+1, \ldots, n\}} \omega_j^{(n)} P_{H_n^+}(w_n) - w_n \right), \]

\[ \mu_n \in [0, 2M_n], \text{ and} \]

\[ M_n := \begin{cases} 
\frac{\sum_{j \in \{n-q+1, \ldots, n\}} \omega_j^{(n)} \| P_{H_n^+}(w_n) - w_n \|^2}{\| \sum_{j \in \{n-q+1, \ldots, n\}} \omega_j^{(n)} P_{H_n^+}(w_n) - w_n \|^2}, & \text{if } w_n \notin \bigcap_{j \in \{n-q+1, \ldots, n\}} H_n^+, \\
1, & \text{otherwise}. 
\end{cases} \]
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 $f : \mathbb{R}^m \rightarrow \mathbb{R}$. Assume there exists a kernel function $\kappa(\cdot, \cdot) : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}$ such that
Reproducing Kernel Hilbert Spaces (RKHS)

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

\[ \kappa(x, \cdot) \in \mathcal{H}, \quad \forall x \in \mathbb{R}^m, \]

Sergios Theodoridis (Uni of Athens)  Adaptive Processing and Projections  January 16, 2009  28 / 56
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

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

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

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

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

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

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

Properties
**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

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

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

**Properties**

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

---

Sergios Theodoridis (Uni of Athens)

Adaptive Processing and Projections

January 16, 2009 28 / 56
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

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

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

Properties

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

Sergios Theodoridis (Uni of Athens)  
Adaptive Processing and Projections  
January 16, 2009 28 / 56
The Goal

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

\[ x_n \mapsto \kappa(x_n, \cdot), \]
The Goal

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

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

\[
y_n(f(x_n) + b) = y_n(\langle f, \kappa(x_n, \cdot) \rangle + b) \geq 0, \quad \forall n.
\]
Find all those $f$ so that $\langle f, y_n \kappa(x_n, \cdot) \rangle \geq 0$, $n = 0, 1, \ldots$
Find all those \( f \) so that \( \langle f, y_n \kappa(x_n, \cdot) \rangle \geq 0, \quad n = 0, 1, \ldots \)

The Equivalence Set

\[
H^+_n := \{ f \in \mathcal{H} : \langle f, y_n \kappa(x_n, \cdot) \rangle \geq 0 \}, \quad n = 0, 1, \ldots
\]
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 $f_0 \in \mathcal{H}$,

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

where the extrapolation coefficient $\mu_n \in [0, 2M_n]$ with

$$M_n := \begin{cases} \frac{\sum_{j \in \mathcal{J}_n} \omega_j^{(n)} \|P_{H_j^+}(f_n) - f_n\|^2}{\| \sum_{j \in \mathcal{J}_n} \omega_j^{(n)} P_{H_j^+}(f_n) - f_n \|^2}, & \text{if } f_n \notin \bigcap_{j \in \mathcal{J}_n} H_j^+, \\ 1, & \text{otherwise.} \end{cases}$$
By mathematical induction on the previous algorithmic procedure, for each index \( n \), there exist \((\gamma_i^{(n)})\) such that

\[
f_n := \sum_{i=0}^{n-1} \gamma_i^{(n)} \kappa(x_i, \cdot).
\]
Sparsification

Recall that as time goes by:

\[ f_n := \sum_{i=0}^{n-1} \gamma^{(n)}_i \kappa(\mathbf{x}_i, \cdot). \]
Recall that as time goes by:

\[ f_n := \sum_{i=0}^{n-1} \gamma_i^{(n)} \kappa(x_i, \cdot). \]

Memory and computational load grows unbounded as \( n \to \infty \)!
Sparsification

Recall that as time goes by:

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

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]:

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

Recall that as time goes by:

\[ f_n := \sum_{i=0}^{n-1} \gamma_i^{(n)} \kappa(x_i, \cdot). \]

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]:

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

Goal

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

\[ f \in \mathcal{B} \cap \left( \bigcap_{n} H_n^+ \right). \]
\[ f_{n+1} := P_B \left( f_n + \mu_n \left( \sum_{j \in J_n} \omega_j^{(n)} P_{H_j}^+(f_n) - f_n \right) \right), \quad \forall n \in \mathbb{Z}_{\geq 0}. \]

\[ \mu_n \in [0, 2M_n], \quad M_n \geq 1, \]
Geometric Illustration of the Algorithm

\[ f_{n+1} := P_B \left( f_n + \mu_n \left( \sum_{j \in \mathcal{I}_n} \omega_j^{(n)} P_{H_j^+}(f_n) - f_n \right) \right), \]

\[ \forall n \in \mathbb{Z}_{\geq 0}. \]

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

\[ f_n \]

\[ f_n \]
\( f_{n+1} := P_B \left( f_n + \mu_n \left( \sum_{j \in \mathcal{J}_n} \omega_j^{(n)} P_{H^+_j} (f_n) - f_n \right) \right), \quad \forall n \in \mathbb{Z}_{\geq 0}. \)

\[ \mu_n \in [0, 2M_n], \quad M_n \geq 1, \]
\[ f_{n+1} := P_B \left( f_n + \mu_n \left( \sum_{j \in J_n} \omega_j^{(n)} P_{H_j^+} (f_n) - f_n \right) \right), \quad \forall n \in \mathbb{Z}_{\geq 0}. \]

\[ \mu_n \in [0, 2M_n], \quad M_n \geq 1, \]
\[ f_{n+1} := P_B \left( f_n + \mu_n \left( \sum_{j \in \mathcal{J}_n} \omega_j^{(n)} P_{H_j^+}(f_n) - f_n \right) \right), \quad \forall n \in \mathbb{Z}_{\geq 0}. \]

\[ \mu_n \in [0, 2M_n], \quad M_n \geq 1, \]
Geometric Illustration of the Algorithm

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

\[ \mu_n \in [0, 2M_n], \quad M_n \geq 1, \]
\[
\begin{align*}
\begin{aligned}
\quad \quad \quad f_{n+1} & := \quad P_B \left( f_n + \mu_n \left( \sum_{j \in J_n} \omega_j^{(n)} \, P_{H_j^+} (f_n - f_n) \right) \right), \\
\quad \quad \quad \mu_n & \in [0, 2M_n], \quad M_n \geq 1,
\end{aligned}
\end{align*}
\]

\[\forall n \in \mathbb{Z}_{\geq 0}.\]
Geometric Illustration of the Algorithm

\[
f_{n+1} := P_B \left( f_n + \mu_n \left( \sum_{j \in J_n} \omega_{j}^{(n)} P_{H_j^+}(f_n) - f_n \right) \right), \quad \forall n \in \mathbb{Z}_{\geq 0}.
\]

\[\mu_n \in [0, 2M_n], \quad M_n \geq 1,\]

**Remark:** It can be shown that this scheme leads to a forgetting factor effect, as in adaptive filtering!
The linear $\epsilon$-insensitive loss function case

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

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

\[
|\langle f, \kappa(x_n, \cdot) \rangle - y_n| \leq \epsilon, \quad \forall n.
\]
The Piece of Information

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

\[
|\langle f, \kappa(x_n, \cdot) \rangle - y_n| \leq \epsilon, \quad \forall n.
\]

The Equivalence Set (Hyperslab)

\[
S_n := \{ f \in \mathcal{H} : |\langle f, \kappa(x_n, \cdot) \rangle - y_n| \leq \epsilon \}, \quad \forall n.
\]
Projection onto a Hyperslab

\[ P_{S_n}(f) = f + \beta \kappa(x_n, \cdot), \forall f \in \mathcal{H}, \]

where

\[ \beta := \begin{cases} 
\frac{y_n - \langle f, \kappa(x_n, \cdot) \rangle - \epsilon}{\kappa(x_n, x_n)}, & \text{if } \langle f, \kappa(x_n, \cdot) \rangle - y_n < -\epsilon, \\
0, & \text{if } |\langle f, \kappa(x_n, \cdot) \rangle - y_n| \leq \epsilon, \\
-\frac{\langle f, \kappa(x_n, \cdot) \rangle - y_n - \epsilon}{\kappa(x_n, x_n)}, & \text{if } \langle f, \kappa(x_n, \cdot) \rangle - y_n > \epsilon.
\end{cases} \]
Projection onto a Hyperslab

\[ P_{S_n}(f) = f + \beta \kappa(x_n, \cdot), \forall f \in \mathcal{H}, \]

where

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

The feasibility set

For each pair \((x_n, y_n)\), form the equivalent hyperslab \(S_n\), and

\[ \text{find } f_\ast \in \bigcap_n S_n. \]
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 $f_0 \in \mathcal{H}$,

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

where the extrapolation coefficient $\mu_n \in [0, 2M_n]$ with

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

\[ S_{n-1} \]

\[ f_n \]

\[ S_n \]
Geometric Illustration of the Algorithm
Geometric Illustration of the Algorithm

\[ S_{n-1} \]

\[ P_{S_{n-1}}(f_n) \]

\[ S_n \]

\[ P_{S_n}(f_n) \]

\[ f_n \]

\[ f_{n+1} \]
Geometric Illustration of the Algorithm

$$f_{n+1} = P_{S_{n-1}}(f_n) + P_{S_n}(f_n)$$
Geometric Illustration of the Algorithm

The diagram illustrates the algorithm with the following notations:

- $S_{n-1}$ and $S_{n+1}$ are two sets.
- $P_{S_{n-1}}(f_n)$ and $P_{S_{n+1}}(f_{n+1})$ are projections onto $S_{n-1}$ and $S_{n+1}$, respectively.
- The algorithm updates the projection as follows:
  
  $$P_{S_n}(f_{n+1}) = P_{S_{n+1}}(f_{n+1}) + f_{n+1} - P_{S_{n-1}}(f_n)$$

This equation shows how the projection changes as the algorithm progresses.
Geometric Illustration of the Algorithm

\[ S_{n-1} \]
\[ S_{n+1} \]
\[ P_{S_{n-1}}(f_n) \]
\[ f_n \]
\[ f_{n+1} \]
\[ f_{n+2} \]
\[ P_{S_n}(f_{n+1}) \]
\[ P_{S_n}(f_n) \]
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 $v \in V$.

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

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

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

Find a point in $K \cap \Pi$, $K$: an icecream cone, $\Pi$: a hyperplane.
Given \((x_n, y_n)\), find an \(f \in \mathcal{H}\) such that [Slavakis, Theodoridis ’07 and ’08]

\[
|\langle f, \kappa(x_n, \cdot) \rangle - y_n| \leq \epsilon
\]

subject to
Given \((x_n, y_n)\), find an \(f \in \mathcal{H}\) such that [Slavakis, Theodoridis ’07 and ’08]

\[
|\langle f, \kappa(x_n, \cdot) \rangle - y_n| \leq \epsilon \quad \text{subject to}
\]

\(f \in V \quad \text{(Affine constraint)}, \quad \text{and} / \text{or}
\]
Given \((x_n, y_n)\), find an \(f \in \mathcal{H}\) such that [Slavakis, Theodoridis ’07 and ’08]

\[
|\langle f, \kappa(x_n, \cdot) \rangle - y_n| \leq \epsilon \quad \text{subject to}
\]

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

\[
\langle f, h \rangle \geq \gamma, \forall h \in B[\tilde{h}, \delta] \quad \text{(Robustness)}.
\]
Algorithm for Robust Regression in RKHS

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

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

where the extrapolation coefficient $\mu_n \in [0, 2M_n]$ with

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

$$f_n := \sum_{l=1}^{L_c} \alpha_i^{(n)} \tilde{h}_l + \sum_{i=0}^{n-1} \gamma_i^{(n)} \kappa(x_i, \cdot), \quad \forall n.$$
Recall that

\[ f_n := \sum_{l=1}^{L_c} \alpha_l^{(n)} \tilde{h}_l + \sum_{i=0}^{n-1} \gamma_i^{(n)} \kappa(x_i, \cdot), \quad \forall n. \]
Recall that

\[ f_n := \sum_{l=1}^{L_c} \alpha_l^{(n)} \tilde{h}_l + \sum_{i=0}^{n-1} \gamma_i^{(n)} \kappa(x_i, \cdot), \quad \forall n. \]

Memory and computational load grows unbounded as \( n \to \infty \)!
Recall that

\[ f_n := \sum_{l=1}^{L_c} \alpha_l^{(n)} \tilde{h}_l + \sum_{i=0}^{n-1} \gamma_i^{(n)} \kappa(x_i, \cdot), \quad \forall n. \]

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

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

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

\[ f_n := \sum_{l=1}^{L_c} \alpha_l^{(n)} \tilde{h}_l + \sum_{i=0}^{n-1} \gamma_i^{(n)} \kappa(x_i, \cdot), \quad \forall n. \]

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

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

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

**Goal**

Thus, we are looking for a classifier \( f \in H \) such that

\[ f \in B \cap K \cap \Pi \cap (\bigcap_{n} S_n). \]
\[ f_n \]
$S_n$

$S_{n-1}$

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

\[ P_{S_{n-1}}(f_n) \]

\[ S_n \]

\[ S_{n-1} \]

\[ K \]

\[ P_K(f'_n) \]
$$f_n - 1_S \Pi$$

$$P_{S_n}(f_n)$$

$$P_{K}(f_n')$$

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

$$f_n$$

$$f_n'$$

$$S_n$$

$$S_{n-1}$$

$$K$$

$$\Pi$$
\[ S_n(P_{\Pi P_K(f'_n)}) = P_{\Pi P_K(f'_n)} \]

\[ K(\Pi f'_n) = P_{\Pi P_K(f'_n)} \]

\[ P_{\Pi P_K(f'_n)} \]

\[ P_{S_n}(f_n) \]

\[ P_{S_{n-1}}(f_n) \]

\[ f_n \]

\[ f'_n \]

\[ S_n \]

\[ S_{n-1} \]
\[ P_{\Pi} P_{K}(f'_{n}) \]
The quadratic $\epsilon$-insensitive loss function case

\[ \Theta_n(f) := \max\{0, (\langle f, \kappa(x_n, \cdot) \rangle - y_n)^2 - \epsilon\}, \quad \forall f \in \mathcal{H}, \forall n. \]
The quadratic $\epsilon$-insensitive loss function case

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

Piece of Information: $C_n := \{ f \in \mathcal{H} : \Theta_n(f) \leq 0 \}.$
The quadratic $\epsilon$-insensitive loss function case

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

Piece of Information: $C_n := \{ f \in \mathcal{H} : \Theta_n(f) \leq 0 \}.$
The quadratic $\epsilon$-insensitive loss function case

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

**Piece of Information:** $C_n := \{ f \in \mathcal{H} : \Theta_n(f) \leq 0 \}.$
The quadratic $\epsilon$-insensitive loss function case

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

Piece of Information: $C_n := \{f \in \mathcal{H} : \Theta_n(f) \leq 0\}.$
The quadratic $\epsilon$-insensitive loss function case

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

**Piece of Information:** $C_n := \{ f \in \mathcal{H} : \Theta_n(f) \leq 0 \}.$
The quadratic $\epsilon$-insensitive loss function case

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

Piece of Information: $C_n := \{ f \in \mathcal{H} : \Theta_n(f) \leq 0 \}.$
The quadratic $\epsilon$-insensitive loss function case

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

Piece of Information: $C_n := \{ f \in \mathcal{H} : \Theta_n(f) \leq 0 \}.$
The quadratic $\varepsilon$-insensitive loss function case

$$\Theta_n(f) := \max\{0, (\langle f, \kappa(x_n, \cdot) \rangle - y_n)^2 - \varepsilon\}, \quad \forall f \in \mathcal{H}, \forall n.$$ 

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 $\Theta_n$ is non-differentiable the subgradient $\Theta'_n$ 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 $\Theta_n$ is non-differentiable the subgradient $\Theta_n'$ 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

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

Example (Projection Mapping)

![Diagram of projection mapping]

The diagram illustrates the projection mapping $P_C$ from a point $f_1$ in $\mathcal{H}$ to its projection $P_C(f_1)$ on $C$, and similarly for $f_2$ to $P_C(f_2)$. The distances $\|f_1 - f_2\|$ and $\|P_C(f_1) - P_C(f_2)\|$ are shown, indicating the nonexpansive property.
Definition (Subgradient)

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

$$\langle g - f, \Theta'_n(f) \rangle + \Theta_n(f) \leq \Theta_n(g), \forall g \in \mathcal{H}.$$
Given a convex continuous function $\Theta_n$, 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 $\Theta_n$, 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 (Fixed Point Set)

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

Define at $n \geq 0$, $\Omega_n := \text{Fix}(T) \cap (\arg \min_{f \in \mathcal{H}} \Theta_n(f))$. 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 $\text{ri}_\Pi(\Omega) \neq \emptyset$. Then, there exists a $f_* \in \text{Fix}(T)$ such that $\lim_{n \to \infty} f_n =: f_*$.

- **Characterization of the limit point.** Assume that $\text{int}(\Omega) \neq \emptyset$. Then, the limit point
  $$f_* \in \text{clos}\left(\liminf_{n \to \infty} \Omega_n\right),$$
  where $\liminf_{n \to \infty} \Omega_n := \bigcup_{m=0}^\infty \bigcap_{n \geq m} \Omega_n$. 
Adaptive Beamforming in RKHS

Preprocessing

\[ y_n := \Re(b_0(k)) \]
\[ y_{n+1} := \Im(b_0(k)) \]

Beamformer \( f \in \mathcal{H} \)

\[ r(k) := \sum_{l=0}^{J} \alpha_l b_l(k) s_l + n(k), \quad \forall k \geq 0, \quad s_l : \text{Steering vectors.} \]
**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 $\tilde{s}_0$ on the actual SOI steering vector $s_0$ (e.g. imperfect array calibration), find a solution that gives uniform output for all the steering vectors in an area around $\tilde{s}_0$; use a closed ball $B[\tilde{s}_0, \delta]$.

\[ \downarrow \]

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

Constraints: Given erroneous information $\tilde{s}_0$ on the actual SOI steering vector $s_0$ (e.g. imperfect array calibration), find a solution that gives uniform output for all the steering vectors in an area around $\tilde{s}_0$; use a closed ball $B[\tilde{s}_0, \delta]$.

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.
Numerical Results
Beam-Patterns

<table>
<thead>
<tr>
<th></th>
<th>Input</th>
<th>LCMV</th>
<th>KRLS</th>
<th>APSM</th>
</tr>
</thead>
<tbody>
<tr>
<td>SINR (dB)</td>
<td>-23.26</td>
<td>-20.21</td>
<td>Very low</td>
<td>18.65</td>
</tr>
</tbody>
</table>
Numerical Results

Convergence Results

![Graph showing convergence results for different methods

- APSM
- Kernel RLS
- LCMV

Root Mean Squared Distance vs. Number of Training Data]
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.