The Projectron: a Bounded Kernel-Based Perceptron

F. Orabona J. Keshet B. Caputo

Idiap Research Institute
Martigny, Switzerland

Swiss Federal Institute of Technology (EPFL)
Lausanne, Switzerland

International Conference on Machine Learning 2008

Outline

Online Learning with Kernels

- What? – Online algorithms observe examples in a sequence of rounds, and construct the classification function incrementally.
- Why? – Kernel-based discriminative online algorithms have been shown to perform very well on binary classification problems.
- How? – They keep a subset of the instances called *support set*, the classification function is then defined by a kernel-dependent weighted combination of the stored examples.

Online Learning with Kernels

- What? – Online algorithms observe examples in a sequence of rounds, and construct the classification function incrementally.
- Why? – Kernel-based discriminative online algorithms have been shown to perform very well on binary classification problems.
- How? – They keep a subset of the instances called *support set*, the classification function is then defined by a kernel-dependent weighted combination of the stored examples.
- But! – Each time an instance is misclassified it is added to the support set.

Bounded Online Learning with Kernels

- If the data is not linearly separable, the algorithms will never stop updating the classification function. Using kernels it means that the size of the solution will grow forever.
- Sooner or later the online classifier will be too slow to be used.
- It would better to have a maximum size of the solution.

Discarding Old Samples

- The first algorithm to overcome the unlimited growth of the support set was proposed by Crammer et al. (2003). The basic idea is to keep discarding vectors from the solution, once the maximum dimension, has been reached.
- The algorithm has been then refined by Weston et al. (2005).
- A similar strategy has also been used in NORMA (Kivinen et al., 2004) and SILK (Cheng et al., 2007).

Discard & Bounds

- The previous algorithms do not quantify the damage done by the removal of a sample.
- First algorithms with a fixed budget and a mistake bound:
  - Forgetron (Dekel at al., 2006)
  - Random Bugdet Perceptron (RBP) (Cesa-Bianchi et al., 2006)
- Both discard samples from the support set to keep the size of the solution constrained by a fixed budget.

Discarding is the Only Way?

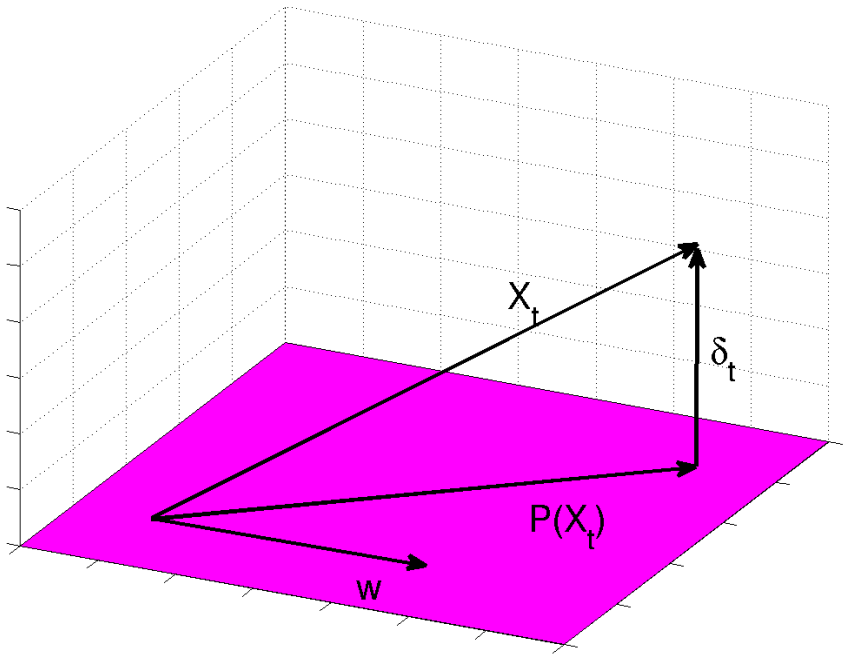
- Discarding is “damaging” the current solution.
- We can only hope to reduce the damage.
- All these algorithms are based on the Perceptron.
- Is there another way to bound the size of the support set and to have a good mistake bound?

Let's Start from the Linear Perceptron

for $t = 1, 2, \ldots, T$ **do**
  Receive new instance $\mathbf{x}_t$
  Predict $\hat{y}_t = \text{sign}(\langle \mathbf{w}, \mathbf{x}_t \rangle)$
  Receive label y_t
  if $y_t \neq \hat{y}_t$ **then**
    $\mathbf{w} = \mathbf{w} + y_t \mathbf{x}_t$
    Add $\mathbf{x}_t$ to the support set
  end if
end for

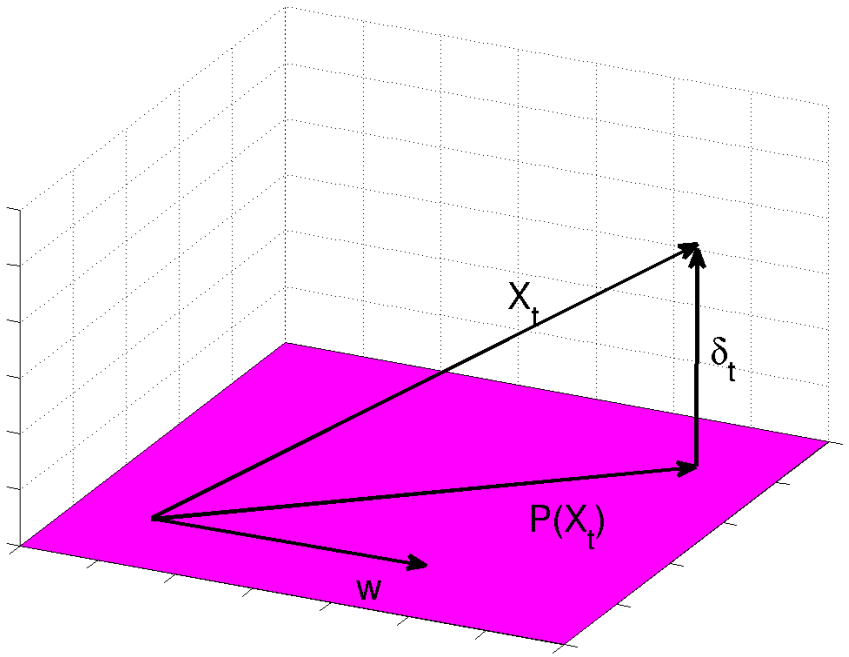
- Fix $\mathbf{u}$ an an arbitrary vector in $\mathcal{H}$
- $\ell(\mathbf{u}, \mathbf{x}_t, y_t) := \max\{0, 1 - y_t \langle \mathbf{u}, \mathbf{x}_t \rangle\}$
- Let $(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_T, y_T)$ be a sequence of instance-label pairs where $\mathbf{x}_t \in \mathcal{X}$, $y_t \in \{-1, +1\}$, and $\|\mathbf{x}_t\| \leq 1$ for all t.
- The number of mistakes of the Perceptron is bounded by $\|\mathbf{u}\|^2 + 2 \sum_i^T \ell(\mathbf{u}, \mathbf{x}_i, y_i)$

Linear Kernel and Projections



- Hence we use the projection of $\mathbf{x}_t$, $P(\mathbf{x}_t)$, for the update. δ_t is the error vector.
- If we have a finite dimensional space and the samples in the support set span it, δ_t will be 0!

Linear Kernel and Projections



- Hence we use the projection of $\mathbf{x}_t$, $P(\mathbf{x}_t)$, for the update. δ_t is the error vector.
- If we have a finite dimensional space and the samples in the support set span it, δ_t will be 0!
- Idea: Instead of discarding old samples from the support set, we project the new ones onto the space spanned by the previous ones.

The Projectron Algorithm

for $t = 1, 2, \ldots, T$ **do**
  Receive new instance $\mathbf{x}_t$
  Predict $\hat{y}_t = \text{sign}(\langle \mathbf{w}, \mathbf{x}_t \rangle)$
  Receive label y_t
  if $y_t \neq \hat{y}_t$ **then**
    $\mathbf{w}' = \mathbf{w} + y_t \mathbf{x}_t$
    $\mathbf{w}'' = \mathbf{w} + y_t P(\mathbf{x}_t)$
    if $\|\delta_t\| = \|\mathbf{w}'' - \mathbf{w}'\| \leq \eta$ **then**
      $\mathbf{w} = \mathbf{w}''$
    else
      $\mathbf{w} = \mathbf{w}'$
      Add $\mathbf{x}_t$ to the support set
    end if
  end if
end for

Note: it is possible to calculate the projection even using Kernels.

What is the Effect of η?

- If $\eta = 0$ we recover the Perceptron algorithm, but we could obtain sparser solutions.
- In the general case of $\eta > 0$ the projection step introduces an error, but it will also give us a smaller solution.
- We want to quantify the effect of different settings of η on the size of the support set and on the performance.

The Support Set Size is Always Bounded!

- If the kernel is finite dimensional it is trivial to show that the maximum size of the support set is finite. However such result holds also for infinite dimensional kernel iff $\eta > 0$ (Engel et al., 2004)
- As opposed to the budget algorithms we do not specify a budget, we just fix η. This will result in a maximum size of the support set.

Theorem (of Boundedness of the Projectron)

Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a continuous Mercer kernel, with $\mathcal{X}$ a compact subset of a Banach space. Then, for any training sequence $(\mathbf{x}_i, y_i), i = 1, \cdots, \infty$ and for any $\eta > 0$, the size of the support set of the Projectron algorithm is finite.

Mistake Bound

What is the influence of η on the performance of the algorithm?

Theorem

Let $(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_T, y_T)$ *be a sequence of instance-label pairs where* $\mathbf{x}_t \in \mathcal{X}$, $y_t \in \{-1, +1\}$, *and* $k(\mathbf{x}_t, \mathbf{x}_t) \leq 1$ *for all* t. *Let* g *be an arbitrary function in* $\mathcal{H}$. *Assume that the Projectron algorithm is run with* $0 \leq \eta < \frac{1}{2\|g\|}$. *Then the number of prediction mistakes the Projectron makes on the sequence is at most*

$$\frac{\|g\|^2 + 2\sum_i^T \ell(g(\mathbf{x}_i), y_i)}{1 - 2\eta\|g\|}$$

Mistake Bound

Theorem (Mistake bound in relation to U)

Let $(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_T, y_T)$ be a sequence of instance-label pairs where $\mathbf{x}_t \in \mathcal{X}$, $y_t \in \{-1, +1\}$, and $k(\mathbf{x}_t, \mathbf{x}_t) \leq 1$ for all t. Let g be an arbitrary function in $\mathcal{H}$, whose norm $\|g\|$ is bounded by U. If the Projectron is run with a parameter η in each round equal to

$$\left(2\ell(f_{t-1}(\mathbf{x}_t), y_t) - \|P_{t-1}k(\mathbf{x}_t, \cdot)\|^2 - 0.5\right)/(2U) \ .$$

Then, the number of prediction mistakes the Projectron makes on the sequence is at most

$$2\|g\|^2 + 4\sum_{i}^{T} \ell(g(\mathbf{x}_i), y_i)$$

Going beyond the Perceptron: the Projectron++

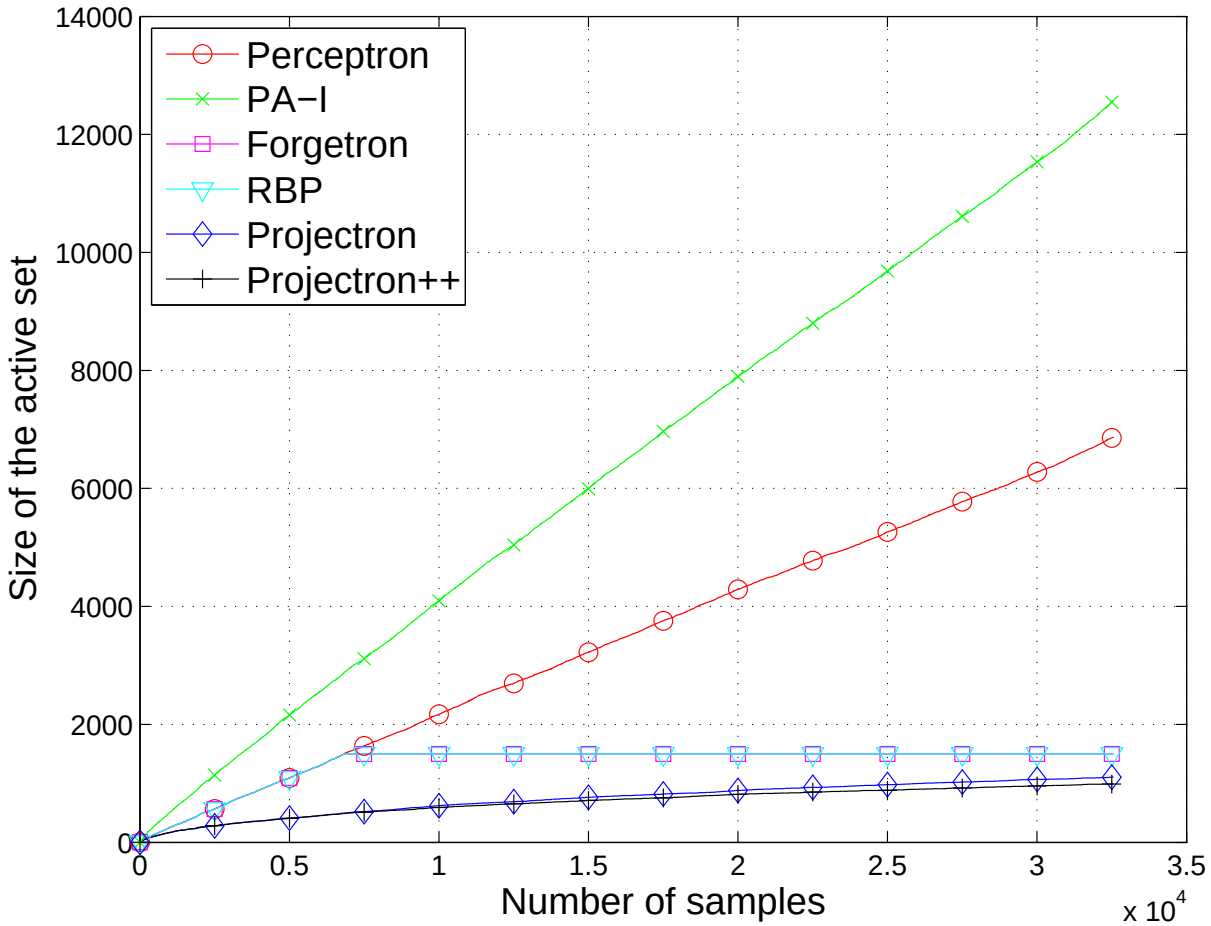
- We can update the hypothesis f also on rounds in which $0 < y_t f(\mathbf{x}_t) < 1$, but only if a projection *is possible*, that is only if the mistake bound is improved by the update.
- If the update would harm the bound we do not perform the update, like in the Perceptron.
- In this way we are adding a margin to the Perceptron algorithm, but only on some updates.

Comparison

- We compared Projectron(++) with Perceptron, PA-I, Forgetron, RBP, and “Stoptron”.
- We set U in Projectron(++) and Forgetron to be the same.
  - This results in a certain budget size for the Forgetron and in an unpredictable size of the solution for the Projectron(++).

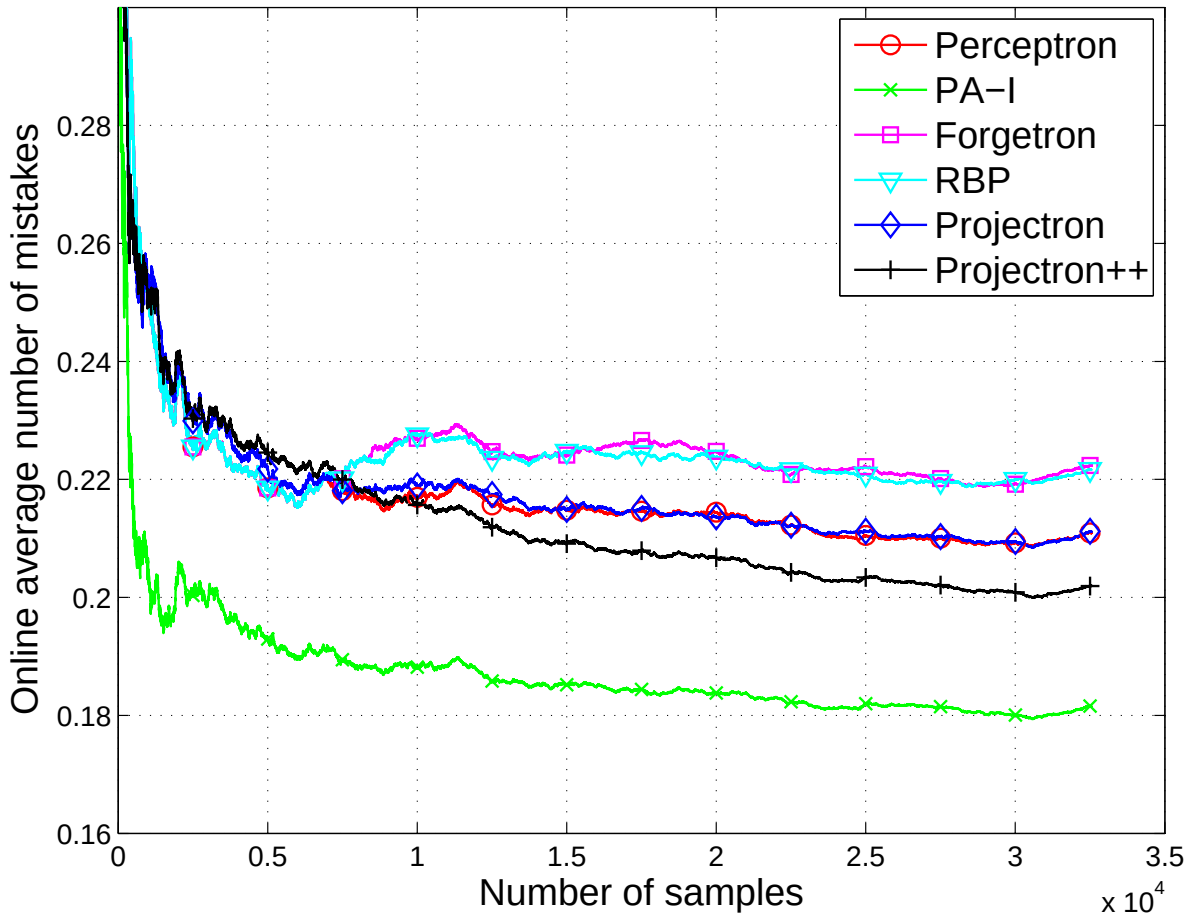
Size of the Support Set

Adult9 dataset, 32561 samples.



Average Online Error

Adult9 dataset, 32561 samples.



Some Numerical Results

Table: Vehicle dataset, 78823 samples.

ALGORITHM	% MISTAKES	SIZE SUPPORT SET
PERCEPTRON	$19.58\% \pm 0.09$	15432.0 ± 69.62
PA-I	$15.27\% \pm 0.05$	30131.4 ± 21.07
	B=4000	
PROJECTRON	$19.63\% \pm 0.08$	3496.4 ± 18.39
PROJECTRON++	$\mathbf{18.27\% \pm 0.06}$	$\mathbf{3187.0 \pm 13.64}$
FORGETRON	$20.40\% \pm 0.04$	4000
RBP	$20.32\% \pm 0.04$	4000
STOPTRON	$19.49\% \pm 3.56$	4000
	B=8000	
PROJECTRON	$19.62\% \pm 0.04$	4668.2 ± 32.88
PROJECTRON++	$\mathbf{18.53\% \pm 0.07}$	$\mathbf{4309.6 \pm 28.67}$
FORGETRON	$19.98\% \pm 0.06$	8000
RBP	$19.94\% \pm 0.06$	8000
STOPTRON	$20.17\% \pm 2.03$	8000

Summary

- This paper presented two different versions of a bounded online learning algorithm. They depend on a parameter that allows to trade accuracy for sparseness of the solution.
- Compared to budget algorithms they have the advantage of a bounded support set size without discarding instances. This keeps performance high.
- Outlook: Although the size of the solution is guaranteed to be bounded, it cannot be determined in advance, and it is not fixed.
- Future work: Mix budget strategy with projection.

Thanks for you attention