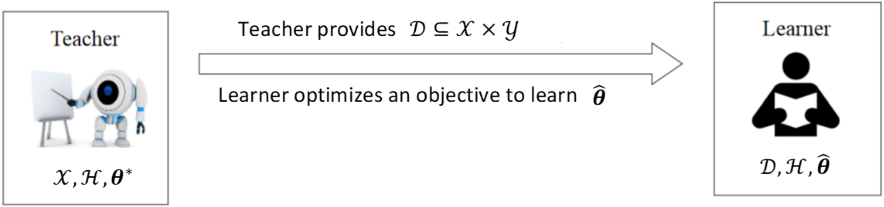
The Teaching Dimension of Kernel Perceptron

Akash Kumar[⋆], Hanqi Zhang[†], Adish Singla[⋆], Yuxin Chen[†]

[⋆]Max Planck Institute for Software Systems [†]University of Chicago

Algorithmic Teaching of ERM Learners

- A helpful teacher to provide labelled inputs to a learner: $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{Y}$
- An ERM (*expected risk minimizer*) learner which receives $\mathcal{D}$ and minimizes a given loss function, e.g., SVM, logistic regression, perceptron.



Teacher

$\mathcal{X}, \mathcal{H}, \boldsymbol{\theta}^*$

Teacher provides $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{Y}$

Learner optimizes an objective to learn $\widehat{\boldsymbol{\theta}}$

Learner

$\mathcal{D}, \mathcal{H}, \widehat{\boldsymbol{\theta}}$

Prior work in the version space setting, where the learner maintains the set of hypothesis consistent with the training examples!

1D Threshold Function: Canonical Example

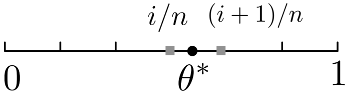


Figure: $f_{\theta^*}(x) = \mathbb{1}\left\{x - \theta^*\right\}$ for $x \in [0, 1]$

For a learner with a finite (or countable infinite) version space, e.g., $\theta \in \{\frac{i}{n}\}_{i=0,\ldots,n}$ where $n \in \mathbb{Z}^+$ (see above), a smallest training set is $\left\{\left(\frac{i}{n}, 0\right), \left(\frac{i+1}{n}, 1\right)\right\}$ where $\frac{i}{n} \le \theta^* < \frac{i+1}{n}$; thus the teaching dimension is 2.

But in the continuous setting, the teaching dimension becomes infinity (∞).

Teaching a 1D Threshold Function

This issue arises because of two key (limiting) modeling assumptions of the version-space learner:

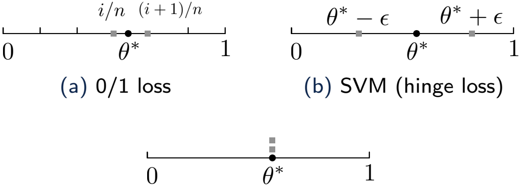
- all (consistent) hypotheses in the version space are treated equally.
- (realizable assumption) there exists a hypothesis in the version space that is consistent with all training examples.

This fails to capture many modern learning algorithms, where the best hypotheses are often selected via optimizing certain loss functions, and the data is not perfectly separable (i.e. not realizable w.r.t. the hypothesis/model class).

Teaching a 1D Threshold Function to an ERM learner

But the problem becomes tractable when the learner is an *empirical risk minimizer* (ERM).

$$\min_{\hat{\boldsymbol{\theta}} \in \mathcal{H}} \mathop{\mathbb{E}}_{(\mathrm{x},y)\sim\mathcal{P}} \left[\ell(f_{\hat{\boldsymbol{\theta}}}(\mathrm{x}), y)\right]$$



(a) 0/1 loss

(b) SVM (hinge loss)

(c) Perceptron

Figure: Teaching a 1D threshold function to an ERM learner. Training instances are marked in grey. (a) Version-space learner with a finite hypothesis set. (b) SVM and training set $\{(\theta^* - \epsilon, 0), (\theta^* + \epsilon, 1)\}$. (c) ERM learner with (perceptron) loss and training set $\{(\theta^*, 0), (\theta^*, 1)\}$.

We note that with these ERM learners, the teaching dimension turns out to be 2 (even in continuous setting).

Teaching in a Constructive Setting

In this work, we study Kernel Perceptron (Non-linear ERM learner) in the contructive setting which extends the work of **Liu and Zhu**, (2016) for teaching linear learners (SVM, Logistic Regression, Ridge Regression).

Constructive setting

Teacher could provide *arbitrarily constructed* training set (teaching examples) $\mathcal{D}$ in the support of the data distribution $\mathcal{P}$.

Kernel Perceptron Learner

Fix the training set $\mathcal{D} := \{(\mathrm{x}_i, y_i)\}_{i=1}^n$ where $\mathrm{x}_i \in \mathbb{R}^d$ and hypothesis $\boldsymbol{\theta} \in \mathbb{R}^d$.

Linear Perceptron

A homogeneous linear perceptron corresponding to hypothesis $\boldsymbol{\theta}$ is defined as:

$$f_{\boldsymbol{\theta}}(\mathrm{x}) := \mathrm{sign}(\boldsymbol{\theta} \cdot \mathrm{x})$$

Perceptron Loss Function

For a labelled point (x, y), hypotheis $\boldsymbol{\theta}$ we consider the perceptron loss ℓ:

$$\ell(f_{\boldsymbol{\theta}}(\mathrm{x}), y) := \max(-y \cdot f_{\boldsymbol{\theta}}(\mathrm{x}), 0)$$

Kernel Perceptron Learner

Optimal Perceptron Algorithm (Learner)

For a given training set $\mathcal{D}$, we consider the optimal perceptron algorithm $\boldsymbol{\mathcal{A}}_{opt}$ which minimizes the perceptron loss as follows:

$$\boldsymbol{\mathcal{A}}_{opt}\left(\mathcal{D}\right) := \mathop{\arg\min}_{\boldsymbol{\theta}\in\mathbb{R}^d} \sum_{i=1}^{n} \ell(f_{\boldsymbol{\theta}}(\mathrm{x}_i), y_i)$$

Kernel Perceptron Learner

Non-Linear Kernel Perceptron Learner

For a kernel operator $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ inducing a *reproducing kernel Hilbert space* (RKHS) $\mathcal{H}_{\mathcal{K}}$, a non-linear kernel perceptron optimizes the following loss:

$$\boldsymbol{\mathcal{A}}_{opt}(\mathcal{D}) := \arg\min_{\boldsymbol{\theta} \in \mathcal{H}_{\mathcal{K}}} \sum_{i=1}^{n} \ell(f_{\boldsymbol{\theta}}(\mathrm{x}_i), y_i)$$

where $f_{\boldsymbol{\theta}}(\cdot) = \sum_{i=1}^{l} \alpha_i \cdot \mathcal{K}(\mathrm{a}_i, \cdot)$ for some $\{\mathrm{a}_i\}_{i=1}^{l} \subset \mathcal{X}$ and α_i real. We also write $f_{\boldsymbol{\theta}}(\cdot) = \boldsymbol{\theta} \cdot \Phi(\cdot)$ where $\Phi : \mathcal{X} \to \mathcal{H}_{\mathcal{K}}$ is defined as feature map to $\mathcal{K}$. A reproducing kernel Hilbert space with $\mathcal{K}$ could be decomposed as $\mathcal{K}(\mathrm{x}, \mathrm{x}') = \langle \Phi(\mathrm{x}),\ \Phi(\mathrm{x}') \rangle$ [?] for any $\mathrm{x}, \mathrm{x}' \in \mathcal{X}$. Thus, we also identify $f_{\boldsymbol{\theta}}$ as $\sum_{i=1}^{l} \alpha_i \cdot \Phi(\mathrm{a}_i)$.

Key Challenges

- Can we exactly teach any non-linear perceptron classifier with a bounded teaching set?
- Is there a trade-off between teaching set size and accuracy to the target classifier?

Contribution (Our Work)

- We formally define approximate teaching of kernel perceptron, and propose a novel measure of teaching complexity, namely the *ϵ-approximate teaching dimension* (ϵ-TD), which captures the complexity of teaching a "relaxed" target that is close to the target hypothesis in terms of the expected risk.

- We establish tight bounds on the teaching dimension of linear and polynomial perceptron. We exhibit optimal training sets that match these teaching dimensions.

- We show that for Gaussian kernelized perceptron, exact teaching is not possible with a finite set of examples, and then establish a $d^{\mathcal{O}\left(\log^2 \frac{1}{\epsilon}\right)}$ bound on the ϵ-approximate teaching dimension.

Contribution (Our Work)

Table 1: Main Results

	linear	**polynomial**	**gaussian**
TD (exact)	$\Theta\left(d\right)$	$\Theta\left(\binom{d+k-1}{k}\right)$	∞
ϵ-approximate TD	-	-	$d^{\mathcal{O}\left(\log^2 \frac{1}{\epsilon}\right)}$
Assumption	-	1	2,3

Teaching Complexity

Fix a kernel perceptron learner, a kernel $\mathcal{K} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ with the corresponding RKHS feature map $\Phi(\cdot)$ and a target model $\boldsymbol{\theta}^* \in \mathcal{H}_{\mathcal{K}}\ \Phi(\cdot)$.

Teaching Set

A set of labelled points $\mathcal{TS} \subseteq \mathcal{X} \times \mathcal{Y}$ provided by a helpful teacher for teaching a target hypothesis $\boldsymbol{\theta}^*$ to a kernel perceptron learner.

Teaching Dimension for Exact Parameters

We define the teaching dimension for *exact* parameter (upto decision boundary) of $\boldsymbol{\theta}^*$ corresponding to a kernel perceptron as $TD(t\boldsymbol{\theta}^*, \boldsymbol{\mathcal{A}}_{opt})$, which is the size of the smallest teaching set $\mathcal{TS}$ such that $\boldsymbol{\mathcal{A}}_{opt}(\mathcal{TS}) = \{t\boldsymbol{\theta}^*\}$ for some real $t > 0$, where

$$TD(\{t\boldsymbol{\theta}^*\}, \boldsymbol{\mathcal{A}}_{opt}) = \min_{\text{real } p>0} TD(p\boldsymbol{\theta}^*, \boldsymbol{\mathcal{A}}_{opt}).$$

Teaching Complexity

Approximate Teaching

Definition (ϵ-approximate TS)

For a given $\epsilon > 0$, we say $\mathcal{TS} \subseteq \mathcal{X} \times \mathcal{Y}$
is an ϵ-approximate teaching set wrt to $\mathcal{P}$ if the kernel perceptron $\hat{\boldsymbol{\theta}} \in \boldsymbol{\mathcal{A}}_{opt}(\mathcal{TS})$ satisfies

$$\left| \mathop{\mathbb{E}}_{(\mathrm{x},y)\sim\mathcal{P}} [\max(-y \cdot f^*(\mathrm{x}), 0)] - \mathop{\mathbb{E}}_{(\mathrm{x},y)\sim\mathcal{P}} \left[\max(-y \cdot \hat{f}(\mathrm{x}), 0)\right] \right| \leq \epsilon$$

where $f^*(\cdot) = \boldsymbol{\theta}^* \cdot \Phi(\cdot),\ \hat{f}(\cdot) = \hat{\boldsymbol{\theta}} \cdot \Phi(\cdot)$

Definition (ϵ-approximate TD)

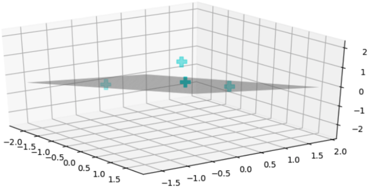
For a given $\epsilon > 0$, we define ϵ-$TD(\boldsymbol{\theta}^*, \boldsymbol{\mathcal{A}}_{opt})$ as the teaching dimension which is the size of the smallest teaching set for ϵ-approximate teaching of $\boldsymbol{\theta}^*$ wrt $\mathcal{P}$.

Teaching Homogeneous Linear Perceptron

We show $TD(\{t\boldsymbol{\theta}^*\}, \boldsymbol{\mathcal{A}}_{opt}) = \Theta(d)$ via constructing a teaching set:

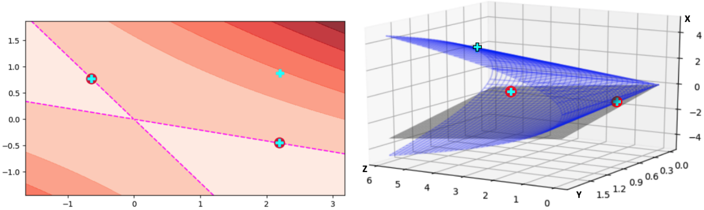
$$\mathrm{x}_i = \mathrm{v}_i, \quad y_i = 1 \quad \forall\, i \;\in\; [d-1];$$

$$\mathrm{x}_d = -\sum_{i=1}^{d-1} \mathrm{v}_i, \quad y_d = 1; \quad \mathrm{x}_{d+1} = \boldsymbol{\theta}^*, \quad y_{d+1} = 1$$



where $\{\mathrm{v}_i\}_{i=1}^{d}$ is an orthogonal basis for $\mathbb{R}^d$ which extends with $\mathrm{v}_d = \boldsymbol{\theta}^*$.

Teaching Non-linear Perceptron



(a) Polynomial ($\mathcal{TS}$)

(b) Polynomial (feature space)

Figure: Numerical examples of exact teaching for polynomial perceptrons. Cyan plus marks and red dots correspond to positive and negative teaching examples respectively.

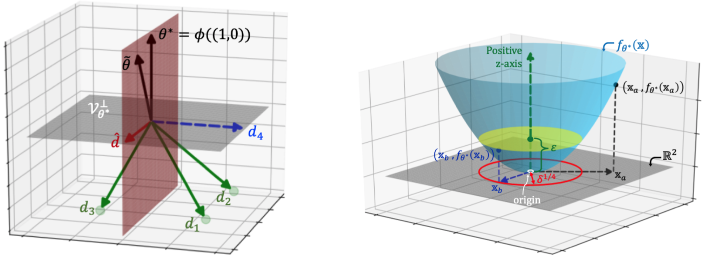
Teaching Non-linear Perceptron

Polynomial Kernel perceptron: underpin a necessary assumption, which leads to limitation of teaching in exact and approximate setting

Assumption

Existence of orthogonal vectors to $\boldsymbol{\theta}^*$ with preimages in original space.

Teaching Non-linear Perceptron



(a)

(b)

Figure: Illustrations for necessity of Assumption (a) Violation of Assumption leads to limitation of Exact teaching ; (b) Pathological case which leads to limitation of Approximate teaching, we demonstrate an example in $\mathbb{R}^2$ with feature space of dimension 3 (where $k = 2$). We consider a model $\boldsymbol{\theta}^* = \frac{1}{\sqrt{2}} \cdot \Phi((1,0)) + \frac{1}{\sqrt{2}} \cdot \Phi((0,1))$. Since k is even, each point $\mathbf{x}$ in $\mathbb{R}^2$ corresponds to a non-negative value of $f_{\theta^*}(\mathbf{x})$.

Teaching Gaussian Kernel Perceptron

- Key Ideas
    - Truncate the taylor features of the Gaussian Kernel to obtain a finite dimensional kernel.
    - Inspired from polynomial setting, we make a necessary assumption on projection of $\boldsymbol{\theta}^*$ in truncated space.
    - Analyze the solutions of perceptron algorithm and pick bounded solutions.

Assumption (Existence of orthogonal classifiers)

Orthogonal vectors in projected space.

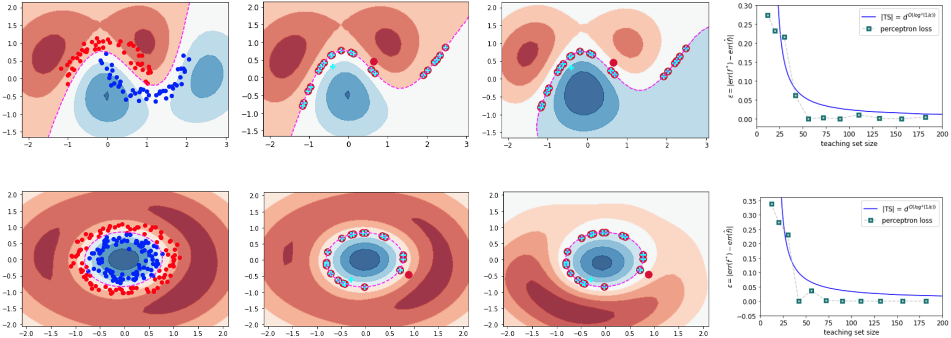
Assumption (Bounded Cone)

The learner optimizes to a bounded solution.

Teaching Gaussian Kernel Perceptron

Main Results

- Under **Assumptions** and $\epsilon > 0$, for any $\hat{f} \in \mathcal{A}_{opt}(\mathcal{TS}_{\boldsymbol{\theta}^*})$ $\left|f^*(\mathrm{x}) - \hat{f}(\mathrm{x})\right| \leq \epsilon$.
- Under **Assumptions** and $\epsilon > 0$, the teaching set $\mathcal{TS}_{\boldsymbol{\theta}^*}$ is an ϵ-approximate teaching set. (Table 1)


$|TS| = d^{O(\log^2(1/\epsilon))}$
perceptron loss
$\epsilon = |err(f^*) - err(\hat{f})|$
teaching set size
$|TS| = d^{O(\log^2(1/\epsilon))}$
perceptron loss
$\epsilon = |err(f^*) - err(\hat{f})|$
teaching set size