# A representation-learning game for classes of prediction tasks

**Neria Uzan, Nir Weinberger**

## Introduction

- The goal is to develop a method for learning data representations that utilizes prior knowledge about the following statistical inference.
- We introduce a game-based formulation for learning dimensionality-reducing representations.

## Game-Theoretic Formulation

$$\text{regret}(R,f\mid P_{\boldsymbol{x}}) := \min_{Q\in\mathcal{Q}_{\mathbb{R}^r}} \mathbb{E}\left[\text{loss}(\boldsymbol{y},Q(R(\boldsymbol{x})))\right] - \min_{Q\in\mathcal{Q}_{\mathcal{X}}} \mathbb{E}\left[\text{loss}(\boldsymbol{y},Q(\boldsymbol{x}))\right]$$

$$\text{regret}_{\text{pure}}(\mathscr{R},\mathscr{F}\mid P_{\boldsymbol{x}}) := \min_{R\in\mathscr{R}}\max_{f\in\mathscr{F}}\mathbb{E}\left[\text{regret}(\boldsymbol{R},f\mid P_{\boldsymbol{x}})\right]$$

$$\text{regret}_{\text{mix}}(\mathscr{R},\mathscr{F}\mid P_{\boldsymbol{x}}) := \min_{\mathsf{L}(\boldsymbol{R})\in\mathscr{P}(\mathscr{R})}\max_{f\in\mathscr{F}}\mathbb{E}\left[\text{regret}(\boldsymbol{R},f\mid P_{\boldsymbol{x}})\right]$$

$$= \max_{\mathsf{L}(f)\in\mathscr{P}(\mathscr{F})}\min_{R\in\mathscr{R}}\mathbb{E}\left[\text{regret}(R,\boldsymbol{f}\mid P_{\boldsymbol{x}})\right]$$

## Theoretical Solution for Linear MSE Setting

- $\mathscr{X}=\mathbb{R}^d$, $\mathscr{Y}=\mathbb{R}$
- $R\in\mathbb{R}^{d\times r}$, $z=R(x)=R^\top x$
- $f\in\mathbb{R}^d$: $\|f\|_S^2\leq 1, S\in\mathbb{S}_{++}^d$
- $Q\in\mathscr{Q}\subset\mathbb{R}^r, Q(z)=q^\top z=R^\top q^\top x\in\mathbb{R}$
- $\boldsymbol{y}=f^\top\boldsymbol{x}+\boldsymbol{n}\in\mathbb{R}$
- $\text{loss}(y_1,y_2)=\|y_1-y_2\|^2$

### Theorem I

$$\text{regret}_{\text{pure}}(\mathscr{F}_S\mid\Sigma_{\boldsymbol{x}})=\lambda_{r+1}\left(\Sigma_{\boldsymbol{x}}^{1/2}S\Sigma_{\boldsymbol{x}}^{1/2}\right)$$

$$R^* := \Sigma_{\boldsymbol{x}}^{-1/2}\cdot V_{1:r}\left(\Sigma_{\boldsymbol{x}}^{1/2}S\Sigma_{\boldsymbol{x}}^{1/2}\right)$$

$$f^* := S^{1/2}\cdot v_{r+1}\left(\Sigma_{\boldsymbol{x}}^{1/2}S\Sigma_{\boldsymbol{x}}^{1/2}\right)$$

### Theorem II

$$\text{regret}_{\text{mix}}(\mathscr{F}_S\mid\Sigma_{\boldsymbol{x}})=\frac{\ell^*-r}{\sum_{i=1}^{\ell^*}\lambda_i^{-1}}$$

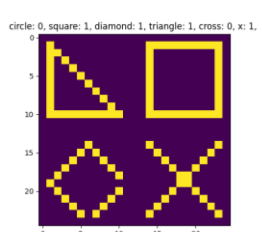
$$\boldsymbol{R}^*=\Sigma_{\boldsymbol{x}}^{-1/2}V_{\mathscr{I}_j}\ w.p.\ p_j$$

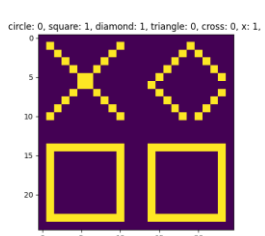
$$\Sigma_f^* := \frac{V^\top\Lambda_{\ell^*}^{-1}V}{\sum_{i=1}^{\ell^*}\lambda_i^{-1}}$$

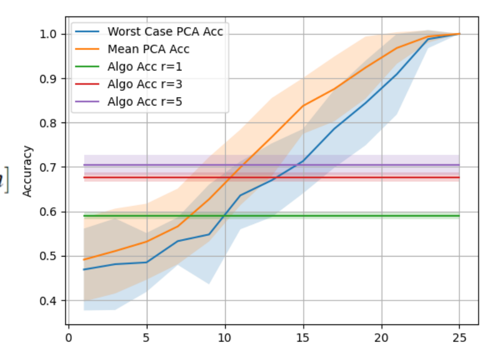
$$\lambda_i\equiv\lambda_i(S^{1/2}\Sigma_{\boldsymbol{x}}S^{1/2}), V\equiv V(S^{1/2}\Sigma_{\boldsymbol{x}}S^{1/2})$$

## Multi Label Setting

- $\mathscr{X}=\mathbb{R}^{\sqrt{d}\times\sqrt{d}}$, that $\mathscr{Y}=\{\pm1\}$
- $R\in\mathscr{R}:=\mathbb{R}^{d\times r}$ where $d>r$
- $\mathscr{Q}:=\{Q(z)=1/[1+\exp(-q^\top z)],\ q\in\mathbb{R}^r\}$
- $\mathscr{F}=\{f_1,\ldots,f_n\}$
- $\text{loss}(y_i,q):=-\frac{1}{2}(1+y_i)\log q-\frac{1}{2}(1-y_i)\log(1-q),\ i\in[n]$




## Algorithm

- Alternating optimization algorithm using gradient descent ascent.
- Finding best representation and worst function at each iteration.
- Resemble the best response algorithm from the field of game theory.

**Algorithm 1** Algorithm I

input $P_{\boldsymbol{x}},\mathscr{R},\mathscr{F},\mathscr{Q},d,r,m,m_0$ ▷ Feature distribution, classes, dimensions and parameters

input $R^{(1)},\{f^{(j)}\}_{j\in[m_0]}$ ▷ Initial representation and initial function (set)

**for** *k=1 to m* **do**

  phase 1: $f^{(m_0+k)}$ is set by a solver of [eq: phase 1 iterative alg] and

$$\text{reg}_k\leftarrow\text{regret}_{\text{mix}}(\{R^{(j)},p^{(j)}\}_{j\in[k]},\mathscr{F}\mid P_{\boldsymbol{x}})$$

  ▷ Solved using Algorithm [alg: phase 1 sol]

  phase 2: $R^{(k+1)},\{p_k^{(j)}\}_{j\in[k+1]}$ is set by a solver of [eq: phase 2 iterative alg] ▷ Solved using Algorithm [alg: phase 2 sol]; step can be removed if k=m

**end**

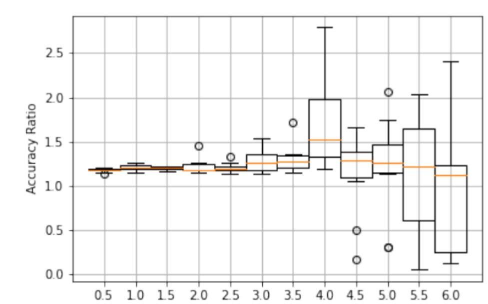
set $m^*=\arg\min_{k\in[m]}\text{reg}_k$

return $\{R^{(j)}\}_{j\in[m^*]} and p_{m_*}=\{p_k^{(j)}\}_{j\in[m^*]}$

## Results

### MSE-Linear

- Comparison between the theoretical results and the results achieved by the algorithm




### Cross Entropy

- Algorithm results for classification
- Regret decrease with m
- Regret increase with d