Learning Eigenvectors for Free

NIPS poster
Granada 12.12.2011

Wouter M. Koolen — Royal Holloway University of London

Wojciech Kotłowski — CWI, Centrum Wiskunde & Informatica

Manfred K. Warmuth — University of California, Santa Cruz

The open problem

Recent interest in matrix generalizations of classical prediction tasks:

- Matrix Hedge (for PCA)
- M. Winnow (learning subspaces)
- M. Exponentiated Gradient (regression)

In each case the matrix generalizations of classical algorithms have performance guarantees (worst-case regret bounds) *identical* to the classical tasks

Symmetric matrices have n^2 parameters and vectors n parameters. Thus matrices should be *harder* to learn!

Are classical bounds loose, or is there a

Free Matrix Lunch?

Our contribution

Extend the classical problem of predicting a sequence of outcomes from finite alphabet to matrix domain

	classical	matrix
outcomes	set of size n	unit vectors in $\mathbb{R}^n$
uncertainty	multinomial	density matrix
parameters	n	n^2

We show how popular online algorithms for learning a multinomial distribution can be extended to learn density matrices

Learning the n^2 parameters of a density matrix should be much harder than learning the n parameters of a multinomial distribution

Surprising, we prove that the worst-case regrets of the classical algorithms and their matrix generalizations are identical:

Free Matrix Lunch!

Algorithms incur no overhead for learning the eigenvectors of the density matrix

Many open problems

- Does the free matrix lunch hold for the matrix minimax algorithm? cf. Shtarkov
- Same questions for other losses
- What properties of the loss function and algorithm cause the free matrix lunch to occur? Proper scoring rules?
- Is there a general regret-bound preserving lift of classical algorithms to matrix prediction?

Probability vector prediction

for trial $t = 1, 2, \ldots$ **do**
  Alg predicts with probability vector $\boldsymbol{\omega}_t$
  Nat returns outcome x_t
  Alg incurs loss $-\log\ \omega_t[x_t]$
end for

Density matrix prediction

for trial $t = 1, 2, \ldots$ **do**
  Alg predicts with density matrix $\boldsymbol{W}_t$
  Nat returns unit vector $\boldsymbol{x}_t$
  Alg incurs loss $-\boldsymbol{x}_t^\top \mathbf{log}(\boldsymbol{W}_t)\boldsymbol{x}_t$
end for

Regret

Loss of the algorithm minus the loss of the best fixed prediction in hindsight
Goal: design online algorithms with low regret

Regret of classical **Laplace** predictor $\boldsymbol{\omega}_{t+1} = \frac{\sum_{q=1}^{t} \boldsymbol{e}_{x_q} + \mathbf{1}}{t+n}$ where $\boldsymbol{e}_i$ is ith basis vector, is

$$\text{Regret}(x_1, \ldots, x_T) \ \leq\ (n-1)\log(T+1)$$

Regret of classical **Krychevsky-Trofimoff** predictor $\boldsymbol{\omega}_{t+1} = \frac{\sum_{q=1}^{t} \boldsymbol{e}_{x_q} + \mathbf{1}/2}{t+n/2}$ is

$$\text{Regret}(x_1, \ldots, x_T) \ \leq\ \frac{n-1}{2}\big(\log(T+1) + \log(\pi)\big)$$

Density matrix

Positive-semidefinite matrix $\boldsymbol{A}$ of unit trace
Decomposition:

$$\boldsymbol{A} \ =\ \sum_i \alpha_i\, \boldsymbol{a}_i \boldsymbol{a}_i^\top$$

where eigenvalues $\boldsymbol{\alpha}$ form probability vector and eigenvectors $\boldsymbol{a}_i$ are orthonormal system

Quantum entropy

$$H(\boldsymbol{A}) \ =\ -\operatorname{tr}(\boldsymbol{A}\,\mathbf{log}\,\boldsymbol{A}),$$

for density matrix $\boldsymbol{A}$ with *matrix logarithm*

$$\mathbf{log}\,\boldsymbol{A} \ =\ \sum_i \log(\alpha_i)\, \boldsymbol{a}_i \boldsymbol{a}_i^\top$$

Equal to Shannon entropy of eigenvalues $\boldsymbol{\alpha}$

Log loss

The log loss is the fundamental loss for **forecasting - data compression - investment**
In matrix case, discrepancy between density matrix prediction $\boldsymbol{W}$ and unit vector outcome $\boldsymbol{x}$ is measured by the *matrix log loss*

$$-\boldsymbol{x}^\top \mathbf{log}(\boldsymbol{W})\boldsymbol{x}$$

Equal to quantum cross entropy

Matrix log loss is proper

The cumulative loss of a fixed prediction $\boldsymbol{W}$

$$\sum_{t=1}^{T} -\boldsymbol{x}_t^\top\, \mathbf{log}(\boldsymbol{W})\boldsymbol{x}_t$$

is minimized at the empirical mean $\boldsymbol{W}^* = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}_t \boldsymbol{x}_t^\top$, with value equal to T times the quantum entropy: $T\, H(\boldsymbol{W}^*)$

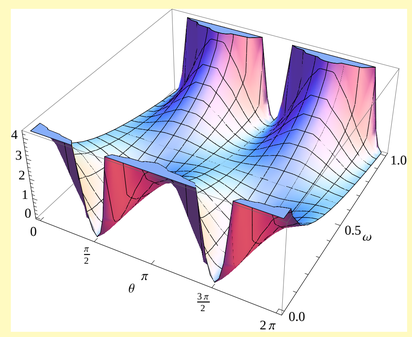
Example: 2D matrix log loss

In $n = 2$ dimensions, we can parametrize the prediction and outcome as follows:

$$\boldsymbol{W} = \begin{pmatrix} \omega & 0 \\ 0 & 1-\omega \end{pmatrix} \quad \text{and} \quad \boldsymbol{x} = \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}$$

with $\omega \in [0,1]$ and $\theta \in [0, 2\pi]$. The loss becomes

$$-\boldsymbol{x}^\top \mathbf{log}(\boldsymbol{W})\boldsymbol{x} \ =\ -\cos^2\theta \log\omega - \sin^2\theta \log(1-\omega)$$



Result: worst-case classical and matrix regret coincide

for both **Matrix Laplace**

$$\boldsymbol{W}_{t+1} \ =\ \operatorname*{argmin}_{\boldsymbol{W}\ \text{dens. mat.}} \left\{ \underbrace{-\operatorname{tr}(\mathbf{log}\,\boldsymbol{W})}_{n\ \text{uniform outcomes}} + \sum_{q=1}^{t} -\boldsymbol{x}_q^\top \mathbf{log}(\boldsymbol{W})\boldsymbol{x}_q \right\} \ =\ \frac{\sum_{q=1}^{t} \boldsymbol{x}_q \boldsymbol{x}_q^\top + \mathbf{I}}{t+n}$$

and **Matrix KT**

$$\boldsymbol{W}_{t+1} \ =\ \operatorname*{argmin}_{\boldsymbol{W}\ \text{dens. mat.}} \left\{ \underbrace{-\frac{1}{2}\operatorname{tr}(\mathbf{log}\,\boldsymbol{W})}_{n/2\ \text{uniform outcomes}} + \sum_{q=1}^{t} -\boldsymbol{x}_q^\top \mathbf{log}(\boldsymbol{W})\boldsymbol{x}_q \right\} \ =\ \frac{\sum_{q=1}^{t} \boldsymbol{x}_q \boldsymbol{x}_q^\top + \mathbf{I}/2}{t+n/2}$$

Any sequence of outcomes not in same eigensystem is suboptimal for Nat