Learning Eigenvectors for Free

Wouter M. Koolen

Royal Holloway
University of London

Wojciech Kotłowski

Politechnika Poznańska

Manfred K. Warmuth

University of California
Santa Cruz

WITMSE 2012

Sunday 30th September, 2012

From learning vectors to learning matrices

- Machine learning is traditionally interested in learning vector parameters (e.g. regression, classification)

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \qquad \boldsymbol{w} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}$$

From learning vectors to learning matrices

- Machine learning is traditionally interested in learning vector parameters (e.g. regression, classification)

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \qquad \boldsymbol{w} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}$$

- Recent interest in matrix generalizations of classical prediction tasks (PCA, learning kernels, learning subspaces)

$$\boldsymbol{X} = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,n} \end{pmatrix} \qquad \boldsymbol{W} = \begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,n} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n,1} & w_{n,2} & \cdots & w_{n,n} \end{pmatrix}$$

The open problem (Warmuth, COLT 2007)

- In each case the matrix generalizations have performance guarantees (worst-case regret bounds) identical to the classical tasks
- Matrices have n^2 parameters and vectors n parameters. Thus matrices should be harder to learn!

The open problem (Warmuth, COLT 2007)

- In each case the matrix generalizations have performance guarantees (worst-case regret bounds) identical to the classical tasks
- Matrices have n^2 parameters and vectors n parameters. Thus matrices should be harder to learn!

Free Matrix Lunch???

This talk

- Predicting n-ary sequence with logarithmic loss
  - Many interpretations: forecasting, data compression, investment
  - Simple but fundamental
  - Extremely well-studied

This talk

- Predicting n-ary sequence with logarithmic loss
  - Many interpretations: forecasting, data compression, investment
  - Simple but fundamental
  - Extremely well-studied
- We generalise the problem and lift the algorithms to the matrix domain.

This talk

- Predicting n-ary sequence with logarithmic loss
  - Many interpretations: forecasting, data compression, investment
  - Simple but fundamental
  - Extremely well-studied
- We generalise the problem and lift the algorithms to the matrix domain.
- We prove and explain a

Free Matrix Lunch!

Outline

Predicting outcomes from individual n-ary sequence (a.k.a. universal coding for n-ary alphabet)

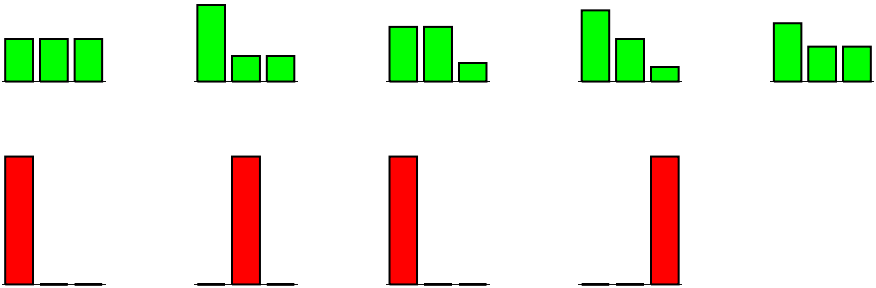
for trial $t = 1, 2, \dots$ **do**
  Alg predicts with a distribution ω_t on n-ary alphabet
  Nat reveals an outcome $x_t \in \{1, \dots, n\}$
  Alg incurs loss $-\log \omega_t(x_t)$
end for

Predicting outcomes from individual n-ary sequence (a.k.a. universal coding for n-ary alphabet)

for trial $t = 1, 2, \ldots$ **do**
  Alg predicts with probability vector (distribution) $\boldsymbol{\omega}_t$
  Nat reveals a basis vector $\boldsymbol{x}_t \in \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}$
  Alg incurs loss $-\log(\boldsymbol{\omega}_t^\top \boldsymbol{x}_t)$
end for

Predicting outcomes from individual n-ary sequence (a.k.a. universal coding for n-ary alphabet)

for trial $t = 1, 2, \dots$ **do**
  Alg predicts with probability vector (distribution) $\boldsymbol{\omega}_t$
  Nat reveals a basis vector $\boldsymbol{x}_t \in \{\boldsymbol{e}_1, \dots, \boldsymbol{e}_n\}$
  Alg incurs loss $-\log(\boldsymbol{\omega}_t^\top \boldsymbol{x}_t)$
end for



Evaluation

- Regret is the cumulative loss of **Alg** minus the loss of the best fixed distribution (prediction):

$$\mathcal{R}_T \;:=\; \sum_{t=1}^{T} -\log\left(\boldsymbol{\omega}_t^\top \boldsymbol{x}_t\right) - \min_{\boldsymbol{\omega}} \sum_{t=1}^{T} -\log\left(\boldsymbol{\omega}^\top \boldsymbol{x}_t\right).$$

Evaluation

- Regret is the cumulative loss of **Alg** minus the loss of the best fixed distribution (prediction):

$$\mathcal{R}_T \;:=\; \sum_{t=1}^{T} -\log\left(\boldsymbol{\omega}_t^\top \boldsymbol{x}_t\right) - \min_{\boldsymbol{\omega}} \sum_{t=1}^{T} -\log\left(\boldsymbol{\omega}^\top \boldsymbol{x}_t\right).$$

- The best distribution $\boldsymbol{\omega}^* = \arg\min \sum_{t=1}^{T} -\log\left(\boldsymbol{\omega}^\top \boldsymbol{x}_t\right)$ is the maximum likelihood estimator, while the loss of $\boldsymbol{\omega}^*$ is the empirical Shannon entropy:

$$\boldsymbol{\omega}^* = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}_t, \qquad \inf_{\boldsymbol{\omega}} \sum_{t=1}^{T} -\log\left(\boldsymbol{\omega}^\top \boldsymbol{x}_t\right) \;=\; T\,H\left(\boldsymbol{\omega}^*\right)$$

- Goal: design online algorithms with low worst-case regret

Algorithms

- Laplace predictor:

$$\boldsymbol{\omega}_{t+1} \;:=\; \frac{\sum_{q=1}^{t} \boldsymbol{x}_q + \mathbf{1}}{t+n} \qquad \mathcal{R}_T \;\leq\; (n-1)\log T + O(1)$$

Algorithms

- Laplace predictor:

$$\boldsymbol{\omega}_{t+1} \;:=\; \frac{\sum_{q=1}^{t} \boldsymbol{x}_q + \mathbf{1}}{t+n} \qquad \mathcal{R}_T \;\le\; (n-1)\log T + O(1)$$

- Krychevsky-Trofimoff (KT) predictor:

$$\boldsymbol{\omega}_{t+1} \;:=\; \frac{\sum_{q=1}^{t} \boldsymbol{x}_q + \mathbf{1}/2}{t+n/2} \qquad \mathcal{R}_T \;\le\; \frac{n-1}{2}\log T + O(1)$$

Algorithms

- Minimax regret achieved by Shtarkov (NML) algorithm:

$$\mathcal{R}_T \;\leq\; \frac{n-1}{2}\log T + O(1)$$

Algorithms

- Minimax regret achieved by Shtarkov (NML) algorithm:

$$\mathcal{R}_T \ \leq \ \frac{n-1}{2}\log T + O(1)$$

- Last Step Minimax algorithm

$$\mathcal{R}_T \ \leq \ \frac{n-1}{2}\log T + O(1)$$

Optimal up to $O(1)$. Beats KT (by a constant).

Outline

Density matrix prediction

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

Density matrix prediction

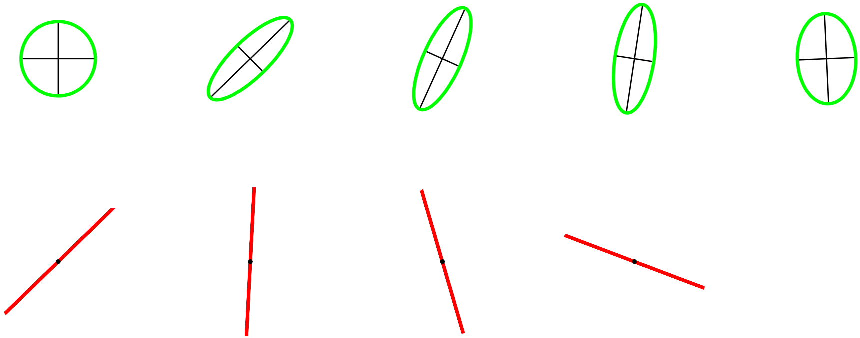
for trial $t = 1, 2, \dots$ **do**
  Alg predicts with density matrix $\boldsymbol{W}_t$
  Nat reveals dyad $\boldsymbol{x}_t \boldsymbol{x}_t^\top$
  Alg incurs loss $-\boldsymbol{x}_t^\top \mathbf{log}(\boldsymbol{W}_t) \boldsymbol{x}_t$
end for

for trial $t = 1, 2, \dots$ **do**
  Alg predicts with distr. $\boldsymbol{\omega}_t$
  Nat reveals $\boldsymbol{x}_t \in \{\boldsymbol{e}_1, \dots, \boldsymbol{e}_n\}$
  Alg incurs loss $-\log(\boldsymbol{\omega}_t^\top \boldsymbol{x}_t)$
end for

Density matrix prediction

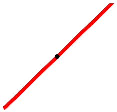
for trial $t = 1, 2, \ldots$ **do**
  Alg predicts with density matrix $\boldsymbol{W}_t$
  Nat reveals dyad $\boldsymbol{x}_t \boldsymbol{x}_t^\top$
  Alg incurs loss $-\boldsymbol{x}_t^\top \mathbf{log}(\boldsymbol{W}_t)\boldsymbol{x}_t$
end for

for trial $t = 1, 2, \ldots$ **do**
  Alg predicts with distr. $\boldsymbol{\omega}_t$
  Nat reveals $\boldsymbol{x}_t \in \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}$
  Alg incurs loss $-\log(\boldsymbol{\omega}_t^\top \boldsymbol{x}_t)$
end for



The outcomes: dyads

- A dyad $\boldsymbol{x}\boldsymbol{x}^\top$ is a rank-one matrix, where $\boldsymbol{x}$ is a vector in $\mathbb{R}^n$ of unit length.



- A dyad is a classical outcome in an arbitrary orthonormal basis:

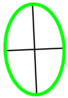
$$\boldsymbol{x}\boldsymbol{x}^\top = \boldsymbol{U}^\top \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \boldsymbol{U}$$

- There are continuously many dyads.

The predictions: density matrices

- A density matrix $\boldsymbol{W}$ is a convex combination of dyads.
  - $\boldsymbol{W}$ is a positive-semidefinite matrix of unit trace.
- A density matrix is a probability vector in an arbitrary orthonormal basis:

$$\boldsymbol{W} \;=\; \sum_{i=1}^{n} \omega_i \, \boldsymbol{a}_i \boldsymbol{a}_i^\top$$



eigenvalues $\boldsymbol{\omega}$	probability vector
eigenvectors $\boldsymbol{a}_i$	orthonormal system

The loss: matrix log loss

- The logarithm of a density matrix $\boldsymbol{W} = \sum_i \omega_i \boldsymbol{a}_i \boldsymbol{a}_i^\top$ is defined by

$$\mathbf{log}(\boldsymbol{W}) \;=\; \sum_i \log(\omega_i)\, \boldsymbol{a}_i \boldsymbol{a}_i^\top.$$

- Discrepancy between prediction $\boldsymbol{W}$ and dyad $\boldsymbol{x}\boldsymbol{x}^\top$: matrix log loss

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

- If **Alg** and **Nat** play in the same eigensystem, i.e. $\boldsymbol{x} = \boldsymbol{a}_j$, then matrix log loss becomes classical log loss:

$$-\boldsymbol{x}^\top \mathbf{log}(\boldsymbol{W})\boldsymbol{x} \;=\; -\boldsymbol{a}_j^\top \sum_i \log(\omega_i)\boldsymbol{a}_i \boldsymbol{a}_i^\top \boldsymbol{a}_j = -\log(\omega_j) = -\log(\boldsymbol{\omega}^\top \boldsymbol{x})$$

Matrix log loss is proper

- The Von Neumann or Quantum entropy:

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

  equals the Shannon entropy of eigenvalues $\boldsymbol{\alpha}$ of $\boldsymbol{A}$.

- We now compete with the empirical Von Neumann entropy:

$$\inf_{\boldsymbol{W}} \sum_{t=1}^{T} -\boldsymbol{x}_t^\top \,\mathbf{log}(\boldsymbol{W})\boldsymbol{x}_t \;=\; T\,H\left(\boldsymbol{W}^*\right) \quad \text{where} \quad \boldsymbol{W}^* \;=\; \frac{\sum_{t=1}^{T} \boldsymbol{x}_t \boldsymbol{x}_t^\top}{T}$$

Outline

Matrix Algorithms

- Matrix Laplace:

$$\boldsymbol{W}_{t+1} \;:=\; \frac{\sum_{q=1}^{t} \boldsymbol{x}_q \boldsymbol{x}_q^\top + \boldsymbol{I}}{t+n}$$

- Matrix Krychevsky-Trofimoff (KT):

$$\boldsymbol{W}_{t+1} \;:=\; \frac{\sum_{q=1}^{t} \boldsymbol{x}_q \boldsymbol{x}_q^\top + \boldsymbol{I}/2}{t+n/2}$$

Matrix Algorithms

- Matrix Laplace:

$$\boldsymbol{W}_{t+1} := \frac{\sum_{q=1}^{t} \boldsymbol{x}_q \boldsymbol{x}_q^\top + \boldsymbol{I}}{t+n} \qquad \boldsymbol{\omega}_{t+1} := \frac{\sum_{q=1}^{t} \boldsymbol{x}_q + \mathbf{1}}{t+n}$$

- Matrix Krychevsky-Trofimoff (KT):

$$\boldsymbol{W}_{t+1} := \frac{\sum_{q=1}^{t} \boldsymbol{x}_q \boldsymbol{x}_q^\top + \boldsymbol{I}/2}{t+n/2} \qquad \boldsymbol{\omega}_{t+1} := \frac{\sum_{q=1}^{t} \boldsymbol{x}_q + \mathbf{1}/2}{t+n/2}$$

Two Free Matrix Lunches

Theorem

Classical and matrix worst-case regrets coincide for Laplace and for KT.

Two Free Matrix Lunches

Theorem

Classical and matrix worst-case regrets coincide for Laplace and for KT.

But why...?

- If **Alg** plays Laplace or KT, then **Nat** will never go out-eigensystem:
  Any sequence of dyads not in same eigensystem is suboptimal for **Nat**
- The classical case is the worst case. No additional regret.
- We learn eigenvectors for free!

Free matrix lunch for Shtarkov?

- Are the classical and matrix prediction games equally hard?
- Ultimate open problem: is the *classical minimax regret*

$$\min_{\boldsymbol{\omega}_1} \max_{\boldsymbol{x}_1} \cdots \min_{\boldsymbol{\omega}_T} \max_{\boldsymbol{x}_T} \sum_{t=1}^{T} - \log\left(\boldsymbol{\omega}_t^\top \boldsymbol{x}_t\right) - T\, H\left(\frac{\sum_{t=1}^T \boldsymbol{x}_t}{T}\right)$$

equal to the *matrix minimax regret*

$$\min_{\boldsymbol{W}_1} \max_{\boldsymbol{x}_1} \cdots \min_{\boldsymbol{W}_T} \max_{\boldsymbol{x}_T} \sum_{t=1}^{T} -\boldsymbol{x}_t^\top \mathbf{log}(\boldsymbol{W}_t)\boldsymbol{x}_t - T\, H\left(\frac{\sum_{t=1}^T \boldsymbol{x}_t \boldsymbol{x}_t^\top}{T}\right)$$

Is there a free matrix lunch for matrix Shtarkov?
- Only numerical evidence for this claim and intermediate conjectures.
- Regret bounds for classical and matrix Last Step Minimax coincide.

Outline

Summary

- Matrix extensions of classical algorithms for log loss.
- Learning a matrix of n^2 parameters with regret for n
- Eigenvectors are learned for free
- Classical data is worst-case

Open questions

- Does the free matrix lunch hold for the matrix minimax algorithm?
- A generic method for promoting classical strategies to the matrix domain.
- Different loss functions.

Thank you!