Lecture 7

Learn "structural information"

- cuts thru lists
- today: permutations

Experts:

- modeled as unit vectors

\[ \text{Expert } i : \mathbf{e}_i = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ 0 \end{pmatrix} \text{ position } i \]

unit vector

- on-line algorithm maintains uncertainty about which expert is best

\[ w_1 \mathbf{e}_1 + w_2 \mathbf{e}_2 + \ldots + w_i \mathbf{e}_i + \ldots + w_m \mathbf{e}_m \]

Probability that the ith expert is best

\[ \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{pmatrix} \]

Uncertainty summarized as probability vector

Experts correspond to corners
PERMUTATIONS

- ORDERING A SET OF OBJECTS
- WANT TO LEARN BEST ORDERING / RANKING ON-LINE

REPRESENTATIONS OF PERMUTATIONS

\[
\begin{pmatrix}
1 & 2 & 3 & 4 \\
2 & 4 & 3 & 1 \\
\end{pmatrix} \quad \Pi : \{1, \ldots, n\}^3 \rightarrow \{1, \ldots, n\}^3
\]

MApMING

\[
\begin{pmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 \\
\end{pmatrix} \quad \text{PERMUTATION}
\]

MAtrIX

\[
L = \begin{pmatrix}
11 & 12 & 13 & 14 \\
21 & 22 & 23 & 24 \\
31 & 32 & 33 & 34 \\
41 & 42 & 43 & 44 \\
\end{pmatrix}
\]

\[
\Pi L = \begin{pmatrix}
21 & 22 & 23 & 24 \\
41 & 42 & 43 & 44 \\
31 & 32 & 33 & 34 \\
11 & 12 & 13 & 14 \\
\end{pmatrix} \quad \text{PERMUTING THE ROWS}
\]

\[
L \Pi = \begin{pmatrix}
12 & 14 & 13 & 11 \\
22 & 24 & 23 & 21 \\
32 & 34 & 33 & 31 \\
42 & 44 & 43 & 41 \\
\end{pmatrix} \quad \text{PERMUTING THE COLUMNS}
\]
\[ m! \text{ PERMUTATIONS} \]
\[ m! \sim n^n e^{-n} \]

**ONE EXPERT PER PERMUTATION**
- TOO EXPENSIVE

2) **OUR LOSS DECOMPOSES INTO A SUM**
- \[ \sum \text{ PRODUCT} \]
- **ALGORITHMS BECOME EFFICIENT**

**MAINTAIN UNCERTAINTY ABOUT PERMUTATION**
AS A MIXTURE OF \( m! \) PERMUTATION MATRICES

\[
\begin{pmatrix}
1 & \vdots & \vdots & \vdots \\
\vdots & 1 & \vdots & \vdots \\
\vdots & \vdots & \ddots & \vdots \\
\vdots & \vdots & \vdots & 1
\end{pmatrix}
\]

\[ = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix} \text{ DETERMINANT} \]

\[ = 0 \text{ ENTRIES} > 0 \]
- **ROw SUMS \& COLUMN SUMS ALL 1**

2) **MAINTAIN DOUBLY STOCHASTIC MATRIX**
- AS PARAMETER
- **IS IT A SUFFICIENT STATISTIC?**
ON-LINE ALGORITHM

- LEARNER CHOOSE $\hat{\pi}$ PROBABILISTICALLY
- BASED ON CURRENT DOUBLY STOCHASTIC MATRIX $W$

- INCR LOSS
  - REPRESENTED AS MATRIX $L \in \mathbb{R}_{+} \times \mathbb{R}_{+}^{m \times n}$

- UPDATE $W$

- $L_{i,j}$ IS LOSS FOR MAPPING $i \to j$

- LOSS OF $\hat{\pi}$

- $\sum_{i} L_{i,\hat{\pi}(i)} = \hat{\pi} \cdot L$

- $L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$
- $\hat{\pi} = \begin{pmatrix} .7 & .5 & .3 \\ 1 & .5 & .3 \\ .9 & 1 & .7 \end{pmatrix}$
- \( \sum_{i} L_{i,\hat{\pi}(i)} = .7 + .5 + .1 = 1.3 \)

\[ W^{t+1} \xrightarrow{\text{LEARNER CHOOSE}} \hat{\pi} \xrightarrow{\text{LEARNER CHOOSE}} \hat{\pi} \]

- LOSS DECOMPOSES
- PRODUCT FORM
- $L_{st}$ SUFFICIENT STATISTIC

COULD IMPLICITLY MAINTAIN THESE

$W^{t+1} \sim \mathcal{E}^{\eta} \sum_{i} L_{i,\hat{\pi}(i)}$
EXPECTED LOSS AT TRIAL $t$

$$E(\hat{\pi})$$

Our parameter matrix will be this expectation

$$W = E(\hat{\pi})$$

\uparrow \text{wrt random choice of alg.}

DOT PRODUCT IS LINEAR:

$$E(\hat{\pi} \cdot L) = E(\hat{\pi}) \cdot L$$

$$= W \cdot L$$

- $W$ current doubly stochastic matrix

- $\hat{\pi}$ chosen probabilistically so that $E(\hat{\pi}) = W$. Details later

- Expected loss $W \cdot L$
Can we capture interesting losses this way?

Typically

\[ L = \prod_{i=1}^{n} M \]

Loss has the form

\[ \uparrow \quad \uparrow \quad \text{loss for identity permutation} \]

\[ \text{correct permutation} \]

\[ \text{permutation} \]

Loss

\[ \text{loss} \left( \hat{\pi}, \pi \right) = \hat{\pi} \cdot \pi \cdot M \]

\[ E \left( \pi \right) = W \cdot \pi \cdot M \]

Examples of loss \( \left( \hat{\pi}, \pi \right) \)

\[ \# \text{ of element } i \text{ where } \hat{\pi}(i) \neq \pi(i) \]

\[ \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} \]

Total distance of elements to correct position

\[ \frac{1}{n-1} \sum_{i=1}^{n} \left| \hat{\pi}(i) - \pi(i) \right| \]

Assure losses are in \[ \text{co} \cdot \pi \]
# OF ELEMENTS MAPPED TO
FIRST HALF BY \( \pi \)
BUT THE SECOND HALF BY \( \hat{\pi} \)

\[
\begin{pmatrix}
0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 \\
1 & 1 & 0 & 0
\end{pmatrix}
\]

# OF ELEMENTS MAPPED TO
FIRST TWO POSITIONS OF \( \pi \)
THAT FAIL TO APPEAR IN
FIRST THREE POSITIONS OF \( \hat{\pi} \)

\[
\begin{pmatrix}
0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

# OF LINKS TRAVERSED
TO FIND 1. ELEMENT OF \( \pi \)
IN LIST ORDERED BY \( \pi \)

\[
\begin{pmatrix}
0 & 1 & 2 & 3 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}
\]

ARE THESE
LOSSSES RICH ENOUGH ?
Algorithm PermELearn: Selecting a permutation

Require: a doubly stochastic $n \times n$ matrix $W$

$A := W$;

for $\ell = 1$ to $n^2 - n + 1$ do

Find permutation $\Pi_\ell$ such that each $A_{i,\Pi_\ell(i)}$ is positive

$\alpha_\ell := \min_i A_{i,\Pi_\ell(i)}$

$A := A - \alpha_\ell \Pi_\ell$

Exit loop if all entries of $A$ are zero

end for \ {at end of loop $W = \sum_{k=1}^{\ell} \alpha_k \Pi_k$}

Randomly select $\Pi_k \in \{\Pi_1, \ldots, \Pi_\ell\}$ using probabilities $\alpha_k$ and return it.

Algorithm 2 PermELearn: Weight Matrix Update

Require: learning rate $\eta$, non-negative loss matrix $L$, and doubly stochastic weight matrix $W$

for each entry $i, j$ of $W$ do

Create $W'$ where each $W'_{i,j} = W_{i,j} e^{-\eta L_{i,j}}$

end for

Create doubly stochastic $\tilde{W}$ by re-scaling the rows and columns of $W'$ (Sinkhorn balancing) and update $W$ to $\tilde{W}$. \ PROJECT ONTO ROW & COLUMN SUM CONSTRAINTS
PERFECT MATCHING

$$A = \begin{pmatrix}
.3 & .2 & .4 \\
.5 & .1 & 0 \\
0 & 0 & .9
\end{pmatrix}$$

PERFECT MATCHINGS

\[ a := \min A_{i, \pi(i)} \]

$$A = A - a \pi$$

\[ \Rightarrow 1 \text{ MORE 0} \]

\[ \leq n^2 - n + 1 \text{ ITERATIONS} \]

\[ \leq n^2 - 2n + 2 \text{ SUFFICE} \]

FOLLOWS FROM PROOF OF BIRKHOFF'S THM
1) DECOMPOSE \( W \)

\[
W = \sum_{k=1}^{n^2-n-1} \alpha_k \pi_k
\]

\( \alpha_k > 0, \sum_k \alpha_k = 1 \)

\( O(n^4) \)

**CHOOSE** \( \pi_k \) **WITH PROBABILITY** \( \alpha_k \)

2) - MULITPLY \( w_{ij} \) **BY FACTOR** \( e^{-\eta L(i,j)} \)

- **DESTROYS ROW & COLUMN NORMALIZATION**

- **RENORMALIZE BY ALTERNATELY NORMALIZING ROW & COLUMN UNTIL CONVERGENCE** (CALLS SINKHORN BALANCING)

**INTERIOR PT METHOD**:

- \( O(n^6 \log \frac{m}{\varepsilon}) \) **TO GET ALL ROWS WITHIN** \( 1 \pm \varepsilon \)

- **MUCH FASTER IN PRACTICE**
SINK HORN BALANCING

\[
\begin{bmatrix}
1 & \frac{4}{3} & 1 \\
\frac{2}{3} & 0 & \frac{2}{3} \\
1 & \frac{2}{3} & 1 \\
1 & \frac{2}{3} & 1
\end{bmatrix}
\]

ROW BAL

\[\downarrow\]

COL BAL

\[
\begin{bmatrix}
\frac{1}{2} & 0 & \frac{1}{2} \\
\frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\
\frac{1}{4} & \frac{1}{2} & \frac{1}{4}
\end{bmatrix}
\]

\[\downarrow\] \(\infty\) STEPS

\[
\begin{bmatrix}
\frac{1}{2} & 0 & \frac{1}{2} \\
\frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\
\frac{1}{4} & \frac{1}{2} & \frac{1}{4}
\end{bmatrix}
\]

\[\uparrow\]

\[\text{DIAGONAL } & \geq 0 \quad \text{DIAGONAL } & \geq 0\]

\[\hat{W} = RW^T\]

\[\uparrow\] NORMALIZED \[\uparrow\] UNNORMALIZED

\[\uparrow\] UNIQUE \[\uparrow\] NOT UNIQUE \[\uparrow\] NOT UNIQUE
ANALYSIS

$$\Delta(A, B) = \sum_{i,j} A_{i,j} \ln \frac{A_{i,j}}{B_{i,j}} + B_{i,j} - A_{i,j}$$

SUM OF UNNORMALIZED RELATIVE ENTROPIES
WHEN SUMMED OVER ROWS (OR COLUMNS):

$$\Delta(A, B) = \sum_{i} \Delta(A_{i*}, B_{i*})$$

$$= \sum_{j} \Delta(A_{*j}, B_{*j})$$

MOTIVATION OF UPDATE

$$\tilde{W} = \text{arg min} \left( \Delta(A, W) + \eta \, A \circ L \right)$$

$$\forall i : |A_{i*}| = 1$$

$$\forall j : |A_{*j}| = 1$$

$$\tilde{W}_{i,j} = \frac{W_{i,j} e^{-\eta L_{i,j}}}{\tilde{W}_{i*} c_{j}}$$

COMBINED UPDATE

WHERE $\gamma_{i}, c_{j}$ CHOSEN SO THAT
CONSTRAINTS SATISFIED

HOW TO ANALYSE
SUCCESSFUL ANALYSIS

SPLIT UPDATE INTO TWO STEPS

1) \[ W' = \text{argmin} \left( \Delta(A, W) + \eta A_{ij} \right) \]
   \[ W'_{ij} = W_{ij} e^{-\eta \Delta_{ij}} \]

2) \[ \tilde{W} = \text{argmin} \Delta(A, W') \]
   \[ \text{row & col sum constr.} \]

- ENOUGH PROGRESS IN 1)
- STEP 2) DOES NOT HURT

WMR IN TWO STEPS (A SIMPLER CASE OF SAME)

\[ W' = \text{argmin} \left( \Delta(\tilde{W}, W) + \eta \tilde{W} \cdot \tilde{E} \right) \]

\[ \tilde{W}'_{i} = W_{i} e^{-\eta \Delta_{i}} \text{ EXPONENTIAL UPDATE} \]

\[ \tilde{W} = \text{argmin} \Delta(\tilde{W}, W') \]

\[ \sum \tilde{W}'_{i} = 1 \]

\[ \tilde{W}_{i} = \frac{\tilde{W}_{i}}{\sum_{j} \tilde{W}_{j}} \text{ NORMALIZATION} \]
\[ \Delta(u, w) - \Delta(u, \tilde{w}) \geq w \cdot l (1 - e^{-\eta}) - \eta u \cdot l \]

In this case, direct proof possible in two steps:

1) \[ \Delta(u, w) - \Delta(u, w') = -\eta u \cdot l + \sum_{i} w_i (1 - e^{-\eta l_i}) \geq w \cdot l (1 - e^{-\eta}) - \eta u \cdot l \]

Enough progress in first step.

2) \[ \Delta(u, w') - \Delta(u, \tilde{w}) \geq 0 \]

\[ \Delta(u, w') = \Delta(u, \tilde{w}) + \Delta(\tilde{w}, w') \]

\[ \Delta(u, w') - \Delta(u, \tilde{w}) = \Delta(\tilde{w}, w') \geq 0 \]
\[ \tilde{w} = \text{argmin} \Delta(\tilde{w}, w') \]
\[ \tilde{w} \in C \]
\[ \uparrow \text{CONVEX} \]

- Holds for arbitrary Bregman divergences (later)

\[ \Delta(u, w') \geq \Delta(u, \tilde{w}) + \Delta(\tilde{w}, w') \]

\( C \) - Convex set

\( \text{projection of } w' \text{ onto } \tilde{w} \)

\( \text{comparator in } C \)
WE CAN NORMALIZE ANY TIME

A ~ B

EQUIVALENT

IF ∃ DIAGONAL R₁(>0): B = RAC

IN THIS CASE:

argmin \( \Delta(\tilde{A}, A) \) = argmin \( \Delta(\tilde{A}, B) \)

cos

sin

ALL EQUIVALENT MATRICES PROJECT TO SAME NORMALIZED MATRIX

OVERALL

- IS THIS ALL USEFUL?

- TECHNIQUES WILL BE USED AGAIN

- DECOMPOSITION OF PARAMETER

- SPLITTING ANALYSIS

UPSHOT:

ENTROPIES CAN BE USED OVER STRUCTURAL DOMAINS

OPEN:

LEARNING ROTATION MATRICES
On-Line Portfolio Selection
Using Multiplicative Updates

David P. Helmbold
UCSC

Robert E. Schapire
AT&T Labs

Yoram Singer
AT&T Labs

Manfred K. Warmuth
UCSC
Constant-Rebalanced Portfolio: A Simple Example

Two Hypothetical Stocks:

No-Pain, No-Gain  \[1,1,1,1,1,1,\ldots\]
Highly Volatile \[\frac{1}{2}, 2, \frac{1}{2}, 2, \frac{1}{2}, 2, \ldots\]

Neither investment alone can increase in value

Rebalancing Strategy: Split the wealth evenly between the two investments, and maintain this even split at the end of each day

On odd days the wealth decreases by:

\[\frac{1}{2} \times 1 + \frac{1}{2} \times \frac{1}{2} = \frac{3}{4}\]

On even days the wealth increases by:

\[\frac{1}{2} \times 1 + \frac{1}{2} \times 2 = \frac{3}{2}\]

After two consecutive days:

\[\frac{3}{4} \times \frac{3}{2} = \frac{9}{8}\]

and it takes six days to double our wealth ...
Definitions

• $N$ stocks

• Relative price change:
  $$\mathbf{x} = (x_1, x_2, \ldots, x_N)$$

• Portfolio vector ($w_i \geq 0$ and $\sum_{i=1}^{N} w_i = 1$):
  $$\mathbf{w} = (w_1, w_2, \ldots, w_N)$$

• Wealth achieved by $\mathbf{w}$:
  $$\mathbf{w} \cdot \mathbf{x} = \sum_{i=1}^{N} w_i x_i$$

• Sequence of price relatives:
  $$\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^T$$

• Sequence of portfolios:
  $$\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^T$$

• Accumulated wealth:
  $$S(\{\mathbf{w}^t\}) = \prod_{t=1}^{T} \mathbf{w}^t \cdot \mathbf{x}^t$$
  $$LS(\{\mathbf{w}^t\}) = \frac{1}{T} \sum_{t=1}^{T} \log (\mathbf{w}^t \cdot \mathbf{x}^t)$$
Constant-Rebalanced Portfolios

\[
S(w) = \prod_{t=1}^{T} w \cdot x^t
\]

\[
LS(w) = \frac{1}{T} \sum_{t=1}^{T} \log (w \cdot x^t)
\]

Best Constant-Rebalanced Portfolio:

\[
w^* \overset{\text{def}}{=} \arg \max_w S(w) = \arg \max_w LS(w)
\]

Universal Portfolios

\{w^t\} is said to be universal if:

\[
\forall \{x^t\} : \lim_{T \to \infty} [LS^* - LS(\{w^t\})] = 0
\]
Cover’s Universal Portfolio Selection Algorithm

A “Bayesian” approach:

\[
    w^t = \frac{\int w \, S_{t-1}(w) \, d\mu(w)}{\int S_{t-1}(w) \, d\mu(w)}
\]

\(d\mu(w)\) is Dirichlet\((1/2, \ldots, 1/2)\) or Dirichlet\((1, \ldots, 1)\)

\[\forall\{x^t\} \quad LS^* - LS(\{w^t\}) = O\left(\frac{N \log(T)}{T}\right)\]

**Update Time:**

Exponential in the number of stocks!
Multiplicative Portfolio Selection Algorithms

Maximize\(^1\) (approximately):

\[
\mathbf{w}^{t+1} = \arg \max_{\mathbf{w}} \eta \log(\mathbf{w} \cdot \mathbf{x}^t) - d(\mathbf{w}, \mathbf{w}^t)
\]

\(d - \) a distance measure (divergence):

\[
D_{RE}(\mathbf{u}||\mathbf{v}) \overset{\text{def}}{=} \sum_{i=1}^{N} u_i \log \frac{u_i}{v_i}
\]

\[
D_{\chi^2}(\mathbf{u}||\mathbf{v}) \overset{\text{def}}{=} \frac{1}{2} \sum_{i=1}^{N} \frac{(u_i - v_i)^2}{v_i}
\]

\(\eta - \) learning rate

\(^1\)Kivinen and Warmuth for on-line regression
Portfolio Update Rules

\( D_{RE} \Rightarrow \text{EG}(\eta)\text{-update:} \)

\[
\omega_{i}^{t+1} = \frac{w_{i}^{t} \exp \left( \eta \frac{x_{i}^{t}}{w_{t}^{t} \cdot x^{t}} \right)}{\sum_{j=1}^{N} w_{j}^{t} \exp \left( \eta \frac{x_{j}^{t}}{w_{t}^{t} \cdot x^{t}} \right)}
\]

\( DK \Rightarrow \chi^{2}(\eta)\text{-update:} \)

\[
\omega_{i}^{t+1} = w_{i}^{t} \left( \eta \left( \frac{x_{i}^{t}}{w_{t}^{t} \cdot x^{t}} - 1 \right) + 1 \right)
\]

Update Time:
Linear in the number of stocks!
Experiments with NYSE Data

30 Stocks, 22-year period

Compared:

• Best single stock
• Best Constant-Rebalanced Portfolio
• Cover’s Universal Portfolio Algorithm
• \(\text{EG}(\eta = 0.05)\) and \(\chi^2(\eta = 0.05)\) updates
<table>
<thead>
<tr>
<th>BCRP</th>
<th>Universal</th>
</tr>
</thead>
<tbody>
<tr>
<td>262.4</td>
<td>98.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EG ($\eta =$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01 0.02 0.05 0.10 0.15 0.20</td>
</tr>
<tr>
<td>119.9 121.5 121.9 113.3 103.1 91.3</td>
</tr>
</tbody>
</table>

Learning Rate
Analysis

**Theorem**
For any sequence $x^1, \ldots, x^T$,

$$LS_T^* - LS_T \leq R\sqrt{\ln N/(2T)}$$

where

$$\forall t : \max_i x_i^t / \min_i x_i^t \leq R$$

$$\eta = 2/R\sqrt{2\log N/T}$$

and $w^1$ is the uniform vector

Weaknesses:
- Need bound on volatility
- Need to know $T$

A ‘Staged’ EG-update overcomes these weaknesses and is a universal portfolio selection algorithm
Variations

• Side (State) Information
• Margin Loans

Research Problems

• Trading Costs:
  – Percentage of the Transactions
  – Fixed Amount Per Transaction
• “Drifting” Portfolios
• ‘Adaptive’ learning rate