CS281B/Stat241B. Statistical Learning Theory. Lecture 13.

Wouter M. Koolen

- Universal Portfolios (See lecture 12 notes)
- Context Tree Weighting

Universal Portfolios

We consider another motivation for competing with the in-hindsight log-optimal portfolio (fixed weights $\boldsymbol{w}$ minimising the mix loss): the stochastic case.

Stochastic portfolio optimization

Suppose price relatives $X_t^k = \frac{\text{price}_{t+1}^k}{\text{price}_t^k}$ are IID.

Expected return of portfolio $\boldsymbol{w}$:

$$\mathbb{E}\left[\prod_{t=1}^{T} \boldsymbol{w}^\intercal \boldsymbol{X}_t\right] \;=\; \left(\mathbb{E}\left[\boldsymbol{w}^\intercal \boldsymbol{X}\right]\right)^T \;=\; \exp\left(T \ln\left(\mathbb{E}\left[\boldsymbol{w}^\intercal \boldsymbol{X}\right]\right)\right)$$

On the other hand, by LLN

$$\frac{1}{T}\sum_{t=1}^{T} \ln\left(\boldsymbol{w}^\intercal \boldsymbol{X}_t\right) \;\to\; \mathbb{E}\left[\ln\left(\boldsymbol{w}^\intercal \boldsymbol{X}\right)\right]$$

so with high probability the return is

$$\prod_{t=1}^{T} \boldsymbol{w}^\intercal \boldsymbol{X}_t \;\approx\; \exp\left(T\,\mathbb{E}\left[\ln\left(\boldsymbol{w}^\intercal \boldsymbol{X}\right)\right]\right)$$

Gap in Jensen amplified exponentially with time T.

Expectation determined by vanishingly rare win streaks: St. Petersburg paradox

Context Tree Weighting

Data compression

In the second half we look at *probability forecasting / data compression*.

Log loss measures code-length of outcome x using "idealised code" P:

$$\text{loss}(P, x) \;=\; -\ln P(x)$$

(Kraft Inequality, Shannon Fano coding, Arithmetic/Range coding)

We will discuss a very powerful and *computationally efficient* learner.

Context

The Picture of Dorian Gray by *Oscar Wilde* from project Gutenberg.

compressor	**size** (KB)
cat	431
gzip	163
7z	137
xz	137
bzip2	121
ppmd	113
ctw	110

Highly competitive results. Online learning/Bayesian methods.

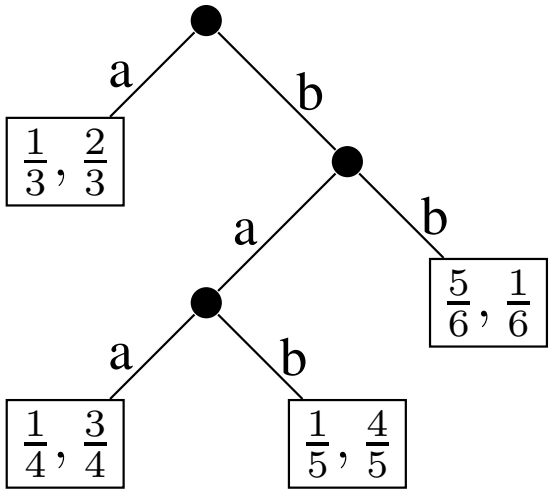
Context Tree

Definition:

$$\mathcal{T} \ ::= \ \triangle(\mathcal{X}) \mid \mathcal{T}^{\mathcal{X}}$$

A *context tree* $\mathcal{T}$ over finite alphabet $\mathcal{X}$ is a rooted full $\mathcal{X}$-ary tree with leaves labelled by probability distributions on $\mathcal{X}$.

Example



Predict next symbol: look up context right-to-left from root, use leaf dist.

$$\overbrace{a\ b\ b\ a\ b}^{\text{context}}\ \underbrace{a\ a\ b}_{\text{used}}\ ? \qquad \Rightarrow \qquad \frac{1}{4}, \frac{3}{4}$$

Do something sensible if context too short:

predict uniform / imaginary default context, ...

Real-world context tree

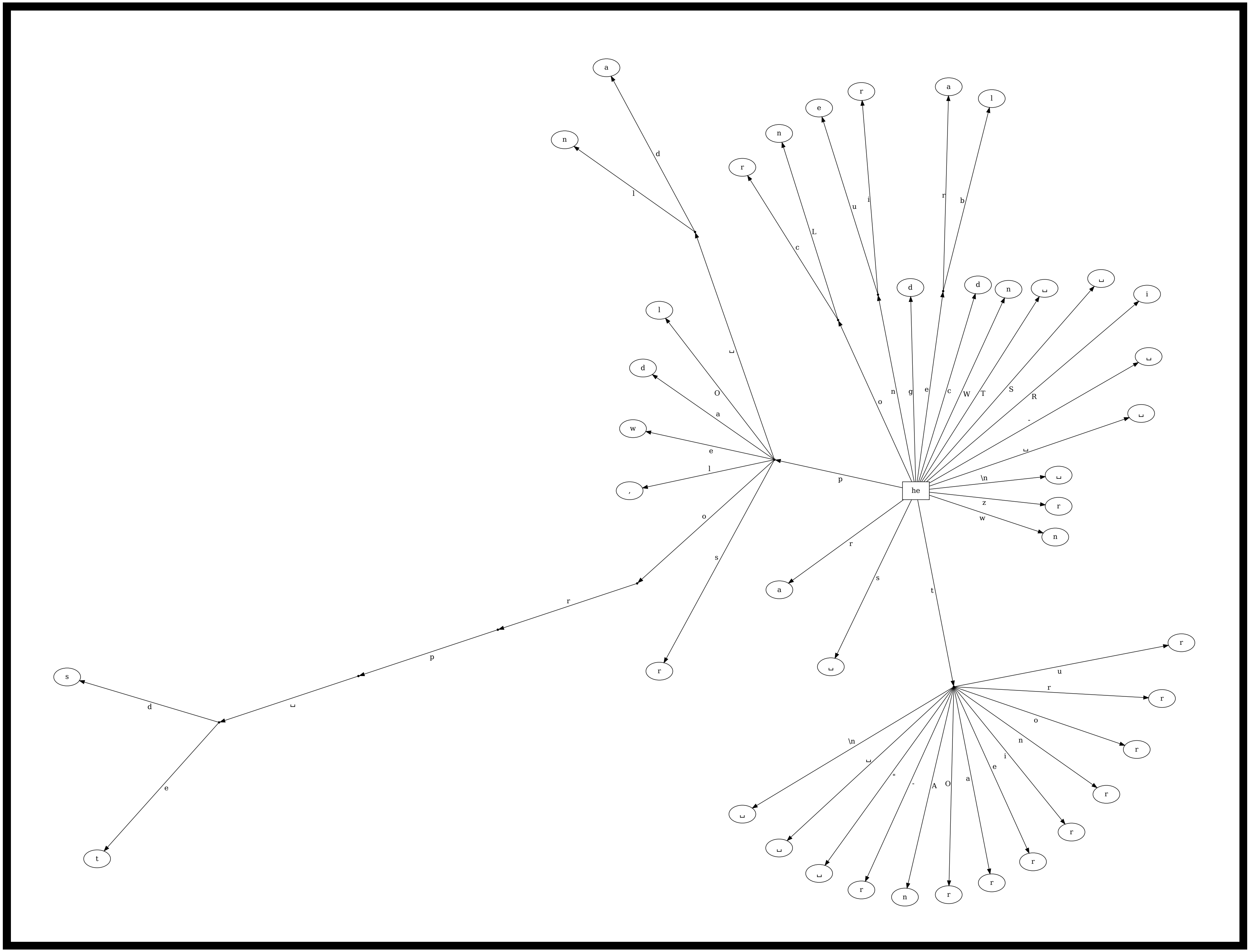
The next slide visualises the MAP context tree on *Picture of Dorian Gray*.

The actual tree is too big, I only show the subtree beneath context “he”.

Leaves are labelled by the most likely successor symbol.

Unused children are omitted.

For example, after “ prophe”, the next symbol is predicted to be “t” if the previous symbol is “e”, while it is predicted to be “s” if the previous symbol was “d”. Indeed the source contains phrases “had prophesied” and “the prophet”.



Goal

Predict/compress as well as the best context tree for the data:

- Learn tree structure
- Learn leaf distributions

Obstacles

Idea: run Aggregating Algorithm on all context trees

Class of context trees is *big*:

- Infinite parametric complexity
- Computation

CTW: use tree structure to implement AA efficiently.

Crucial: independence in prior distribution maintained by update (learning process is local)

Prior

The (recursive) prior process:

- W.p. $\frac{1}{2}$ output leaf, draw dist. $\sim$ Dirichlet$(\frac{1}{2}, \ldots, \frac{1}{2})$.
- W.p. $\frac{1}{2}$ output internal node. Draw $\mathcal{X}$ children independently.

Typically cutoff (force leaf) at some preset maximum depth.

Predict and Update

Prior is Bayesian mixture of recursive "here" vs "deeper".

- "here" integrates out to KT predictor ($\frac{1}{2}$-smoothed frequencies)
- "deeper" is recursive combination

Implementation

Maintain in each node (corresponding to a certain context)

- observation counts of successor symbols
- posterior distribution on "here" vs "deeper"

Predict: follow path from root down corresponding to current context. Aggregate $\frac{1}{2}$-smoothed frequencies (KT) according to posterior distribution. Time $O(d|\mathcal{X}|)$ where d is max tree depth.

Update: follow path from root down corresponding to current context. In each node increment observed symbol count. Update the "here" vs "deeper" weight using Bayes' rule. Time $O(d)$ where d is max tree depth.

Memory: number of nodes bounded by dT, but typically much smaller.

Regret bound

Regret w.r.t. context tree $\mathcal{T}$ (with L leaves, so $\frac{|X|L-1}{L-1}$ nodes total)

$$
\begin{aligned}
-\ln P(x^T) + \ln P_{\mathcal{T}}(x^T) \;&\leq\; \underbrace{\frac{|X|L-1}{L-1}\ln 2}_{\text{tree structure}} + \sum_{\text{leaf } i} \underbrace{\frac{|X|-1}{2}\ln T_i + \ln|X|}_{\text{leaf distribution}} \\
&\leq\; \frac{|X|L-1}{L-1}\ln 2 + L\frac{|X|-1}{2}\ln\frac{T}{L} + L\ln|X|
\end{aligned}
$$

See (Begleiter and El-Yaniv, 2006) for full details.

Discussion

- Maintain posterior probabilities logarithmically.
- Sparsity: Zero-Redundancy-Estimator instead of KT
  smaller regret when only subset of alphabet is used
- 8 "phases": trees with context in bytes to predict next bit
  not so clear why this works better, but it sure does wonders
  Understanding this phenomenon / experimenting with alternatives could be your project
- Path compression (memory efficiency)
  from some point context becomes unique $\Rightarrow$ special tree node
- Best tree complexity grows with $T \Rightarrow$ catch-up-phenomenon $\Rightarrow$ switching