Fast generation of context tree models

Arnaldo Mandel

IME-USP

Antonio Galves

Florencia Leonardi

others . . .

Making sense of sequencial observations

Phenomenon: stochastic process emitting symbols from a finite set, discrete time

abaabcabcccbacbaa

Purpose: model giving good predictions

Making sense of sequencial observations

Phenomenon: stochastic process emitting symbols from a finite set, discrete time

abaabcabcccbacbaa

Purpose: model giving good predictions

The model

A *stochastic process* $\mathbf{X}_n, n \in \mathbb{Z}$ with values in a finite *alphabet* A.
Specified by the *true probabilities*

$$\mathbb{P}(\mathbf{X}_t = x_t \mid \text{all past})$$

How much past must be remembered?

The model

A *stochastic process* $\mathbf{X}_n$, $n \in \mathbb{Z}$ with values in a finite *alphabet* A.
Specified by the *true probabilities*

$$\mathbb{P}(\mathbf{X}_t = x_t \mid \text{all past})$$

How much past must be remembered?

The model

A *stochastic process* $\mathbf{X}_n, n \in \mathbb{Z}$ with values in a finite *alphabet* A.

Specified by the *true probabilities*

$$\mathbb{P}(\mathbf{X}_t = x_t \mid \text{all past})$$

How much past must be remembered?

Context trees

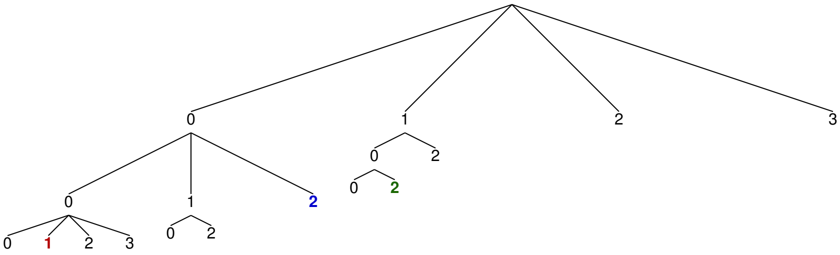
Given a digital tree:

Words are read on paths from nodes to the root:

1 0 0 **2 0** **2 0 1**

Context trees

Given a digital tree:



Words are read on paths from nodes to the root:

1 0 0 **2 0** **2 0 1**

Context trees

Context tree (τ, p):

τ : leaves of a digital tree

$p = \{p(\cdot|w) : w \in \tau\}$: probability distributions on A.

$p(a|w) =$ probability of a appearing after context w

It really models...

Stationary processes: probabilities are fixed on time.

Equivalently: for each length, there is a fixed probability distribution on words of that length.

It really models...

Stationary processes: probabilities are fixed on time.

Equivalently: for each length, there is a fixed probability distribution on words of that length.

How good it is

Likelyhood of a sequence, given a tree: probabilty of occurrence

likelyhood bad — log-likelyhood good!

log-loss = -(log-likelyhood) even better

How good it is

Likelyhood of a sequence, given a tree: probabilty of occurrence

likelyhood bad — log-likelyhood good!

log-loss = -(log-likelyhood) even better

How good it is

Likelyhood of a sequence, given a tree:
probabilty of occurrence

likelyhood bad — log-likelyhood good!

log-loss = -(log-likelyhood) even better

How to find a context tree

Given a sufficiently long observation, relative frequency of words is a maximum likelyhood estimator for probabilities (prayer or ergodicity)

a*cbaa*acba*cbaacbaa*c

$$p(cbaa) = 3/15$$

$=$ number of occurrences of cbaa / number of words of length 4

How to find a context tree

Given a sufficiently long observation, relative frequency of words is a maximum likelyhood estimator for probabilities (prayer or ergodicity)

a*cbaa*acba*cbaacbaa*c

$$p(cbaa) = 3/15$$
$= $ number of occurrences of cbaa / number of words of length 4

General method

Start with the complete digital tree (up to some depth) and prune to taste.

How much?

Contending goals

- Predict the past: the bigger the tree, the better – maximizes likelyhood
- Predict the future: avoid overfitting, parsimony

General method

Start with the complete digital tree (up to some depth) and prune to taste.

How much?

Contending goals

- Predict the past: the bigger the tree, the better – maximizes likelyhood
- Predict the future: avoid overfitting, parsimony

General method

Start with the complete digital tree (up to some depth) and prune to taste.

How much?

Contending goals

- Predict the past: the bigger the tree, the better – maximizes likelyhood
- Predict the future: avoid overfitting, parsimony

General method

Start with the complete digital tree (up to some depth) and prune to taste.

How much?

Contending goals

- Predict the past: the bigger the tree, the better – maximizes likelyhood
- Predict the future: avoid overfitting, parsimony

General method

Start with the complete digital tree (up to some depth) and prune to taste.

How much?

Contending goals

- Predict the past: the bigger the tree, the better – maximizes likelyhood
- Predict the future: avoid overfitting, parsimony

Methods

Many methods prune the tree by offsetting the log-loss by a size penalty

- MDL (Rissanen)
- BIC (Schwartz)
- KT (Krichevcky-Trofimov)
- SMC (Galves et al.)

penalty proportional to log of the length

Methods

Many methods prune the tree by offsetting the log-loss by a size penalty

- MDL (Rissanen)
- BIC (Schwartz)
- KT (Krichevcky-Trofimov)
- SMC (Galves et al.)

penalty proportional to log of the length

Doing it all over

Assign a *cost* $c(w) > 0$ for each word w, and let
$c(\tau) = \sum_{w \in \tau} c(w)$.

For $\alpha \in \mathbb{R}_+$, let

$$\tau_\alpha(x) = \text{tree minimizing} \quad \ell(\tau, x) + \alpha c(\tau) \log n$$

τ is *champion* if $\tau = \tau_\alpha(x)$ for some α.

Doing it all over

Assign a *cost* $c(w) > 0$ for each word w, and let
$c(\tau) = \sum_{w \in \tau} c(w)$.

For $\alpha \in \mathbb{R}_+$, let

$$\tau_\alpha(x) = \text{tree minimizing} \quad \ell(\tau, x) + \alpha c(\tau) \log n$$

τ is *champion* if $\tau = \tau_\alpha(x)$ for some α.

Why the champions?

- All previous tree selection models choose a champion tree, with $C(w) = 1$.

  They are easy to produce – $\widetilde{O}(n)$ algorithm

  Selection can be done with post processing.

Why the champions?

- All previous tree selection models choose a champion tree, with $C(w) = 1$.

  They are easy to produce – $\widetilde{O}(n)$ algorithm

  Selection can be done with post processing.

Why the champions?

- All previous tree selection models choose a champion tree, with $C(w) = 1$.

  They are easy to produce – $\widetilde{O}(n)$ algorithm

  Selection can be done with post processing.

Connection to network flow

$$\longrightarrow$$