

Reordering a tree according to an order on its leaves and studying the evolution of the idiolect of writers

L. Bulteau¹, P. Gambette¹, O. Seminck²

¹ LIGM, CNRS, Université Gustave Eiffel, France

² Lattice, CNRS & ENS/PSL & Université Sorbonne nouvelle, France

LIGM - 2022-05-17

Introduction

Initial Motivation

Studying the evolution of the idiolect of authors

The idiolect according to Dittmar, 1996

“the **language of the individual**, which because of the acquired habits and the stylistic features of the personality **differs from that of other individuals** and in different life phases shows, as a rule, different or differently weighted”

Initial Motivation

Studying the evolution of the idiolect of authors

The idiolect according to Dittmar, 1996

“the **language of the individual**, which because of the acquired habits and the stylistic features of the personality **differs from that of other individuals** and in different life phases shows, as a rule, different or differently weighted”

The idiolect according to Bloch, 1948

“the **totality of the possible utterances of one speaker** at one time in using a language to interact with one other speaker.”

Initial Motivation

Studying the evolution of the idiolect of authors

The idiolect according to Dittmar, 1996

“the **language of the individual**, which because of the acquired habits and the stylistic features of the personality **differs from that of other individuals** and in different life phases shows, as a rule, different or differently weighted”

The idiolect according to Bloch, 1948

“the **totality of the possible utterances of one speaker** at one time in using a language to interact with one other speaker.”

- ▶ We prefer Bloch’s definition, independent of the notion of **style**, which is linked with aesthetic values and judgements.

Initial Motivation

Studying the evolution of the idiolect of authors

Question

Can we measure and characterise how the idiolect of an author evolves with time?

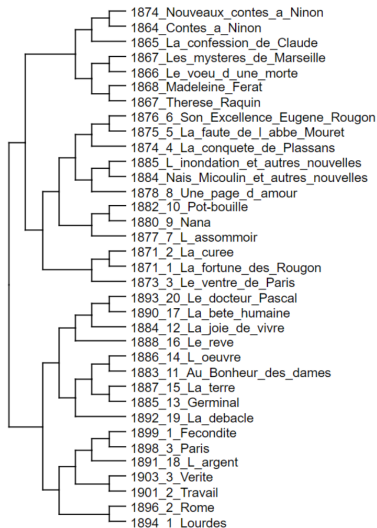
Idiolect project

- ▶ funded by the PR[AI]RIE institute
- ▶ started by Thierry Poibeau, Dominique Legallois and Olga Seminck
- ▶ produced a corpus of novels by 11 prolific 19th century French authors: The Corpus for Idiolectal Research (CIDRE) [Seminck, Gambette, Legallois & Poibeau, *JOHD* 2022]

Initial Motivation

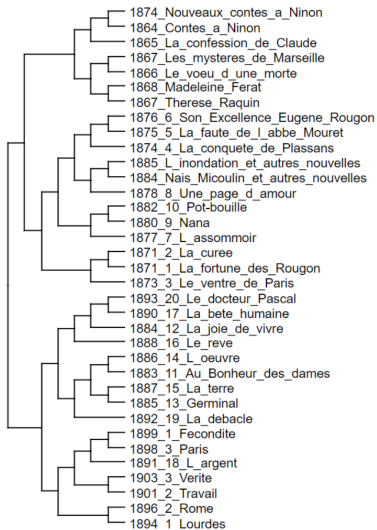
Studying the evolution of the idiolect of authors

- ▶ a natural first step: hierarchical clustering:
 - ▶ compute distances between all pairs of novels of an author, depending on the contents of the novels (linguistic parameters)
 - ▶ perform hierarchical clustering of this distance matrix to get a dendrogram (rooted tree).
- ▶ does the clustering group together novels published in consecutive years?



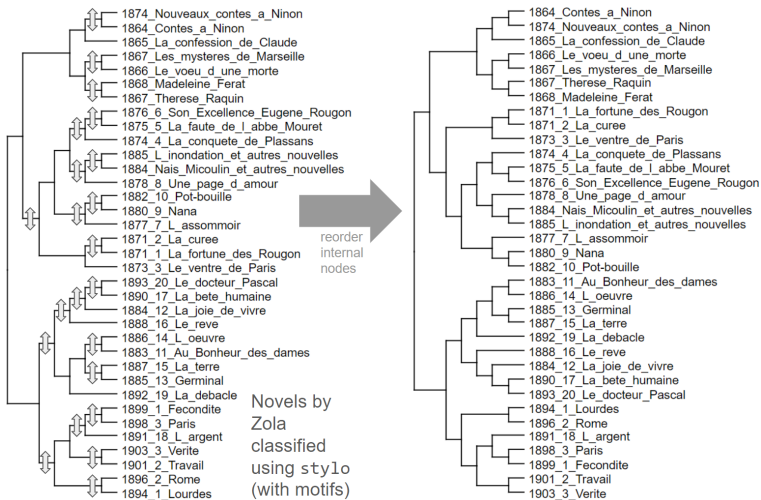
Initial Question

How much is the dendrogram consistent with time?



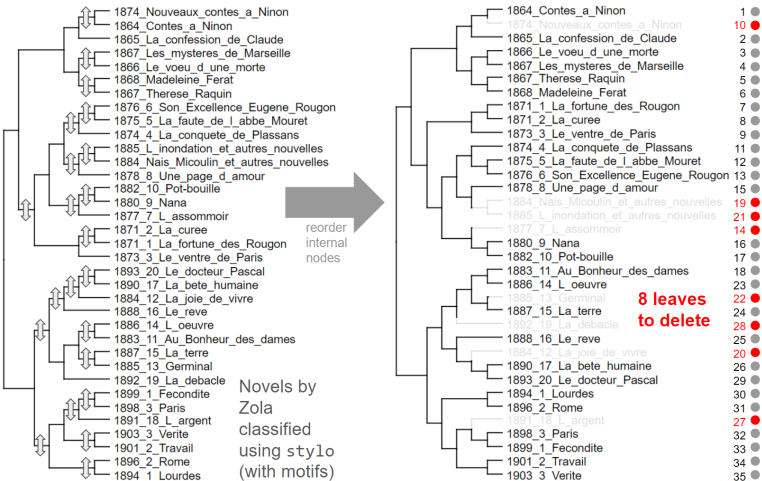
Initial Question

How much is the dendrogram consistent with time?



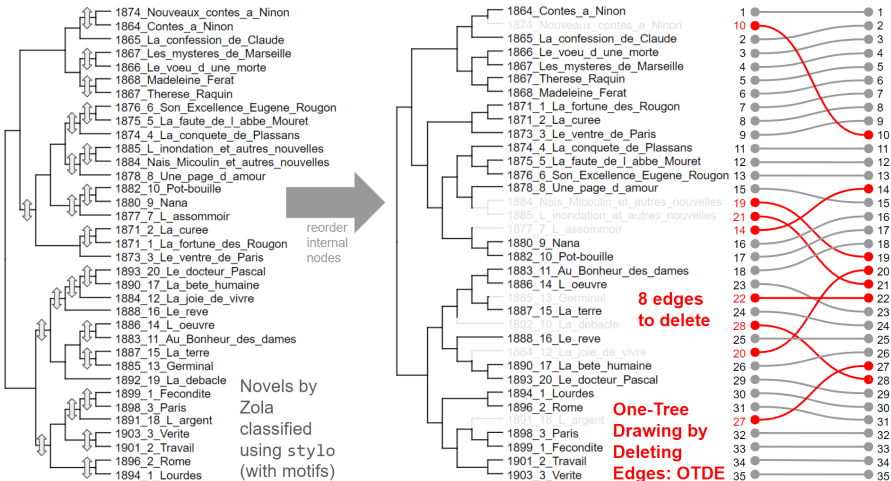
Initial Question

How much is the dendrogram consistent with time?



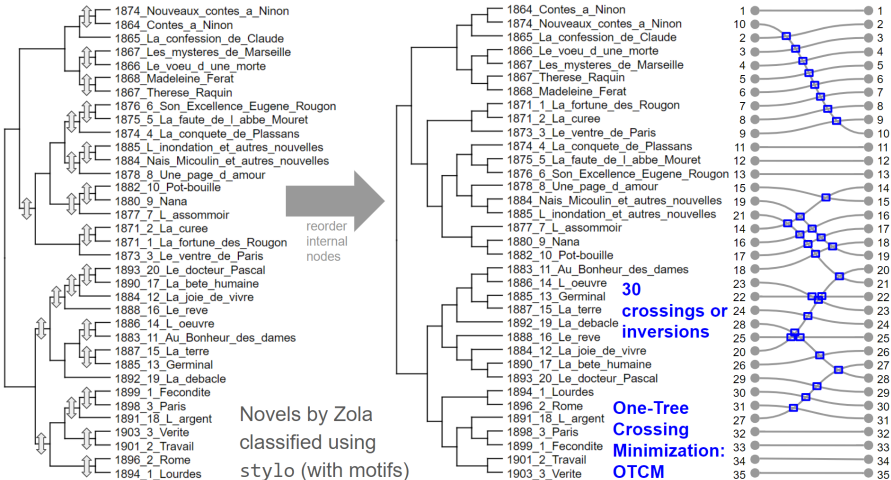
Initial Question

How much is the dendrogram consistent with time?



Initial Question

How much is the dendrogram consistent with time?



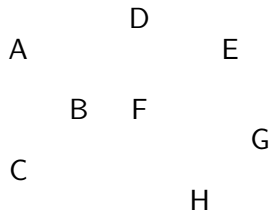
Modelization

Motivation

Is a clustering consistent with external data?

Input:

- ▶ Elements

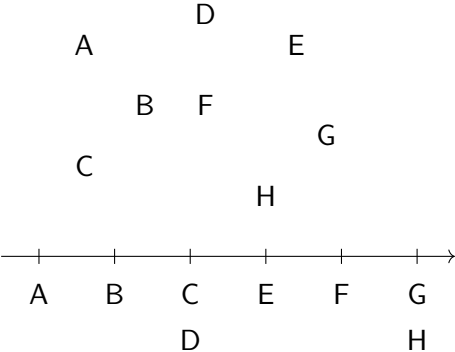


Motivation

Is a clustering consistent with external data?

Input:

- ▶ Elements
- ▶ Ordering (time-line, ...)

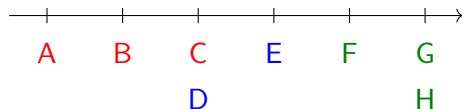
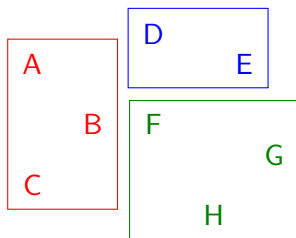


Motivation

Is a clustering consistent with external data?

Input:

- ▶ Elements
- ▶ Ordering (time-line, ...)
- ▶ Clustering



Is the clustering consistent with the ordering?

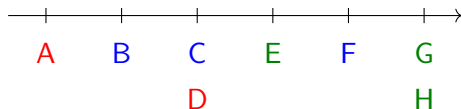
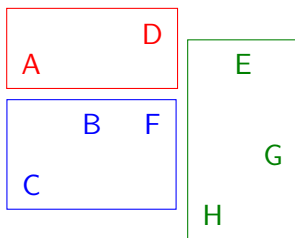


Motivation

Is a clustering consistent with external data?

Input:

- ▶ Elements
- ▶ Ordering (time-line, ...)
- ▶ Clustering



Is the clustering consistent with the ordering?

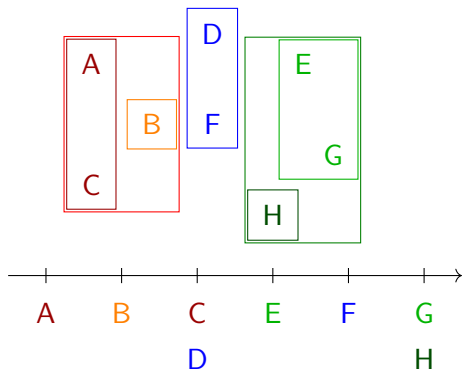


Motivation

Is a clustering consistent with external data?

Input:

- ▶ Elements
- ▶ Ordering (time-line, ...)
- ▶ Hierarchical Clustering



Is the clustering consistent with the ordering?

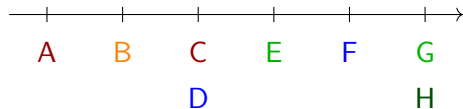
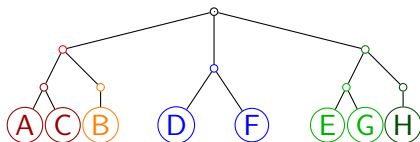


Motivation

Is a clustering consistent with external data?

Input:

- ▶ Elements
- ▶ Ordering (time-line, ...)
- ▶ Hierarchical Clustering (seen as a tree / dendrogram)



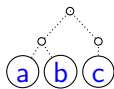
Definitions

- ▶ Tree T with leaf set X , ordering $\sigma: X \rightarrow \mathbb{N}$ (weak order \leq_σ)

Definitions

- ▶ Tree T with leaf set X , ordering $\sigma: X \rightarrow \mathbb{N}$ (weak order \leq_{σ})

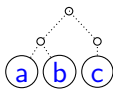
- ▶ Conflict: leaves a, b, c with $a <_{\sigma} c <_{\sigma} b$ and



Definitions

- ▶ Tree T with leaf set X , ordering $\sigma: X \rightarrow \mathbb{N}$ (weak order \leq_σ)

- ▶ Conflict: leaves a, b, c with $a <_\sigma c <_\sigma b$ and



OTDE One-Tree Drawing by Deleting Edges

Given T, σ, k ,

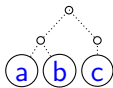
Find $X' \subseteq X, |X'| \geq |X| - k$

Such that $T[X']$ has no conflict with σ

Definitions

- ▶ Tree T with leaf set X , ordering $\sigma: X \rightarrow \mathbb{N}$ (weak order \leq_σ)

- ▶ Conflict: leaves a, b, c with $a <_\sigma c <_\sigma b$ and



- ▶ Ordering of T : strict order σ' without conflict with T

TTDE Two-Tree Drawing by Deleting Edges

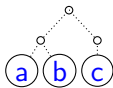
Given T_1 , T_2 , k ,

Find $X' \subseteq X$, $|X'| \geq |X| - k$,

and an ordering σ of both $T_1[X']$ and $T_2[X']$

Definitions

- ▶ Tree T with leaf set X , ordering $\sigma: X \rightarrow \mathbb{N}$ (weak order \leq_σ)



- ▶ Conflict: leaves a, b, c with $a <_\sigma c <_\sigma b$ and
- ▶ Ordering of T : strict order σ' without conflict with T
- ▶ Crossing between σ and σ' : pair $\{a, b\}$ with $a <_\sigma b$ and $b <_{\sigma'} a$

OTCM One-Tree Crossing Minimization

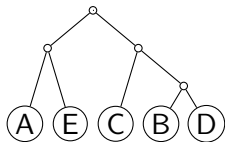
Given T, σ, k ,

Find σ' ordering of T

Such that σ' has at most k crossings with σ

Example

Tree T



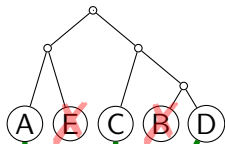
Order σ



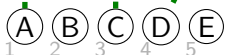
Input instance

Example

Tree T



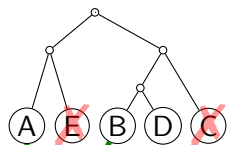
Order σ



Score for OTDE: $k = 2$ deletions

Example

Tree T



Order σ



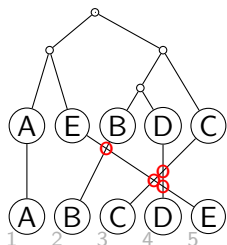
Another solution with the same score

fun fact: all possible permutations of each node's children need 2 deletions

Example

Tree T

Order σ



Score for OTCM: 4 crossings

Previous Results

OTCM on binary trees

Most studied variant, from phylogenetics

- ▶ Dwyer, Schreiber '04: $O(n^2)$
- ▶ Fernau, Kaufmann, Poths '05: $O(n \log^2 n)$
- ▶ Bansal et al. '09: $O(n \log^2 n / \log \log n)$
- ▶ Fernau, Kaufmann, Poths. '10
and Venkatachalam, et al. '10: $O(n \log n)$

Previous Results

OTCM on binary trees

Most studied variant, from phylogenetics

- ▶ Dwyer, Schreiber '04: $O(n^2)$
- ▶ Fernau, Kaufmann, Poths '05: $O(n \log^2 n)$
- ▶ Bansal et al. '09: $O(n \log^2 n / \log \log n)$
- ▶ Fernau, Kaufmann, Poths. '10
and Venkatachalam, et al. '10: $O(n \log n)$

OTDE, TTDE

Introduced by Fernau et al.:

- ▶ Reduction **from** OTDE **to** 3-Hitting Set
- ▶ NP-hardness still open

Our Results

OTCM on arbitrary trees

- ▶ NP-hardness (from Feedback Arc Set)

Our Results

OTCM on arbitrary trees

- ▶ NP-hardness (from Feedback Arc Set)

OTDE

- ▶ NP-hardness (from Independent Set)
- ▶ Parameterized algorithms
 - ▶ (simple) XP for the degree d
 - ▶ (advanced) FPT for the *deletion-degree* ∂ ¹

¹ $\partial = \text{degree of } T[X \setminus X'], \partial \leq \min\{d, k\}$

Our Results

OTCM on arbitrary trees

- ▶ NP-hardness (from Feedback Arc Set)

OTDE

- ▶ NP-hardness (from Independent Set)
- ▶ Parameterized algorithms
 - ▶ (simple) XP for the degree d
 - ▶ (advanced) FPT for the *deletion-degree* ∂ ¹

TTDE

- ▶ NP-hardness (from OTDE)

¹ $\partial = \text{degree of } T[X \setminus X'], \partial \leq \min\{d, k\}$

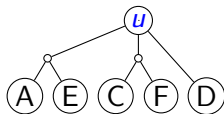
Algorithms

OTDE is XP for the degree

Bottom-up Dynamic Programming

For each internal node v , interval l, r

$X(v, l, r) =$ deletions in $T[v]$ when mapped with $\sigma[l..r]$

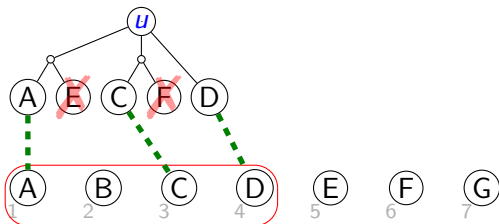


OTDE is XP for the degree

Bottom-up Dynamic Programming

For each internal node v , interval l, r

$X(v, l, r) =$ deletions in $T[v]$ when mapped with $\sigma[l..r]$



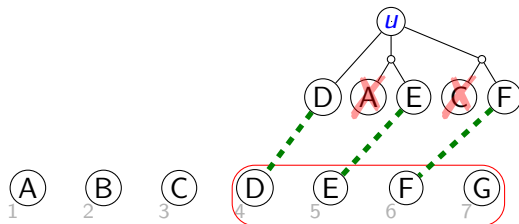
$$X(u, 1, 4) = 2$$

OTDE is XP for the degree

Bottom-up Dynamic Programming

For each internal node v , interval l, r

$X(v, l, r) =$ deletions in $T[v]$ when mapped with $\sigma[l..r]$



$$X(u, 1, 4) = 2$$

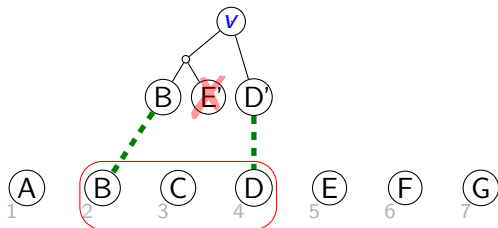
$$X(u, 4, 7) = 2$$

OTDE is XP for the degree

Bottom-up Dynamic Programming

For each internal node v , interval l, r

$X(v, l, r) =$ deletions in $T[v]$ when mapped with $\sigma[l..r]$



$$X(u, 1, 4) = 2$$

$$X(v, 2, 4) = 1$$

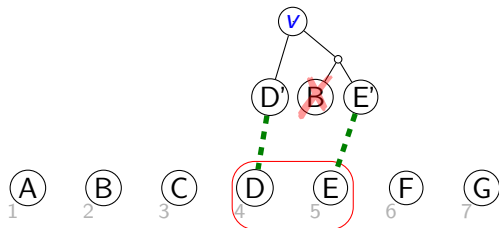
$$X(u, 4, 7) = 2$$

OTDE is XP for the degree

Bottom-up Dynamic Programming

For each internal node v , interval l, r

$X(v, l, r) =$ deletions in $T[v]$ when mapped with $\sigma[l..r]$



$$X(u, 1, 4) = 2$$

$$X(v, 2, 4) = 1$$

$$X(u, 4, 7) = 2$$

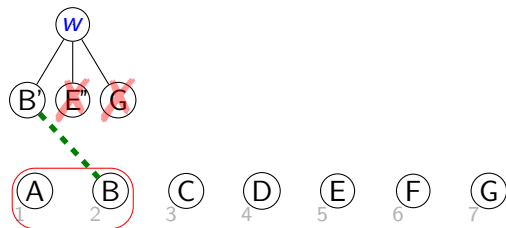
$$X(v, 4, 5) = 1$$

OTDE is XP for the degree

Bottom-up Dynamic Programming

For each internal node v , interval l, r

$X(v, l, r) =$ deletions in $T[v]$ when mapped with $\sigma[l..r]$



$$X(u, 1, 4) = 2$$

$$X(v, 2, 4) = 1$$

$$X(w, 1, 2) = 2$$

$$X(u, 4, 7) = 2$$

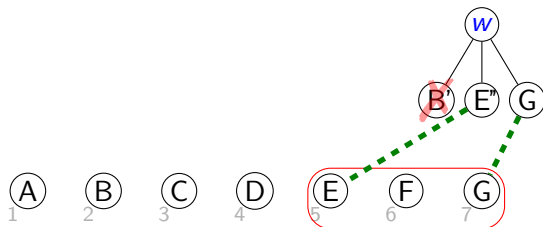
$$X(v, 4, 5) = 1$$

OTDE is XP for the degree

Bottom-up Dynamic Programming

For each internal node v , interval l, r

$X(v, l, r) =$ deletions in $T[v]$ when mapped with $\sigma[l..r]$



$$X(u, 1, 4) = 2$$

$$X(v, 2, 4) = 1$$

$$X(w, 1, 2) = 2$$

$$X(u, 4, 7) = 2$$

$$X(v, 4, 5) = 1$$

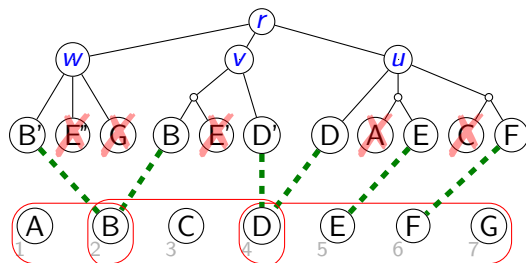
$$X(w, 5, 7) = 1$$

OTDE is XP for the degree

Bottom-up Dynamic Programming

For each internal node v , interval l, r

$X(v, l, r) =$ deletions in $T[v]$ when mapped with $\sigma[l..r]$



$$X(u, 1, 4) = 2$$

$$X(v, 2, 4) = 1$$

$$X(w, 1, 2) = 2$$

$$X(u, 4, 7) = 2$$

$$X(v, 4, 5) = 1$$

$$X(w, 5, 7) = 1$$

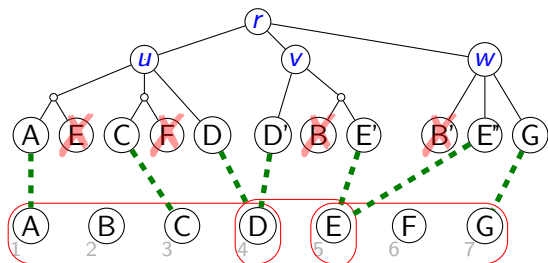
$$X(r, 1, 7) = \min(5, \dots)$$

OTDE is XP for the degree

Bottom-up Dynamic Programming

For each internal node v , interval l, r

$X(v, l, r) =$ deletions in $T[v]$ when mapped with $\sigma[l..r]$



$$X(u, 1, 4) = 2$$

$$X(v, 2, 4) = 1$$

$$X(w, 1, 2) = 2$$

$$X(u, 4, 7) = 2$$

$$X(v, 4, 5) = 1$$

$$X(w, 5, 7) = 1$$

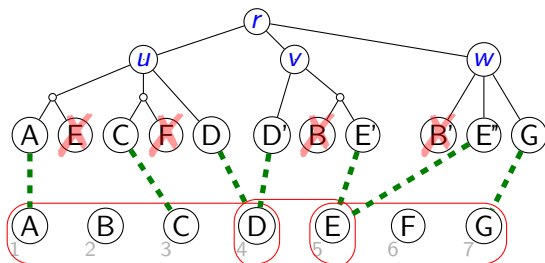
$$X(r, 1, 7) = \min(5, 4, \dots)$$

OTDE is XP for the degree

Bottom-up Dynamic Programming

For each internal node v , interval l, r

$X(v, l, r) =$ deletions in $T[v]$ when mapped with $\sigma[l..r]$



n^3 DP entries

$$X(u, 1, 4) = 2$$

$$X(v, 2, 4) = 1$$

$$X(w, 1, 2) = 2$$

$$X(u, 4, 7) = 2$$

$$X(v, 4, 5) = 1$$

$$X(w, 5, 7) = 1$$

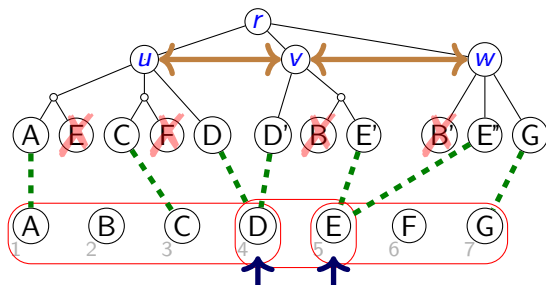
$$X(r, 1, 7) = \min(5, 4, \dots)$$

OTDE is XP for the degree

Bottom-up Dynamic Programming

For each internal node v , interval l, r

$X(v, l, r) =$ deletions in $T[v]$ when mapped with $\sigma[l..r]$



n^3 DP entries

$d!$ permutations
of the children

n^{d-1} pivots

$$X(u, 1, 4) = 2$$

$$X(v, 2, 4) = 1$$

$$X(w, 1, 2) = 2$$

$$X(u, 4, 7) = 2$$

$$X(v, 4, 5) = 1$$

$$X(w, 5, 7) = 1$$

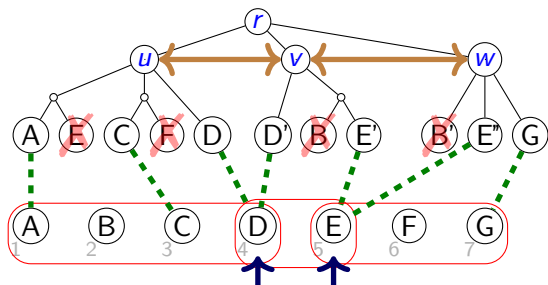
$$X(r, 1, 7) = \min(5, 4, \dots)$$

OTDE is XP for the degree

Bottom-up Dynamic Programming

For each internal node v , interval l, r

$X(v, l, r)$ = deletions in $T[v]$ when mapped with $\sigma[l..r]$



n^3 DP entries

$d!$ permutations
of the children

n^{d-1} pivots

Overall $O(d!n^{d+2})$

$$X(u, 1, 4) = 2$$

$$X(v, 2, 4) = 1$$

$$X(w, 1, 2) = 2$$

$$X(u, 4, 7) = 2$$

$$X(v, 4, 5) = 1$$

$$X(w, 5, 7) = 1$$

$$X(r, 1, 7) = \min(5, 4, \dots)$$

OTDE is FPT for the deletion degree

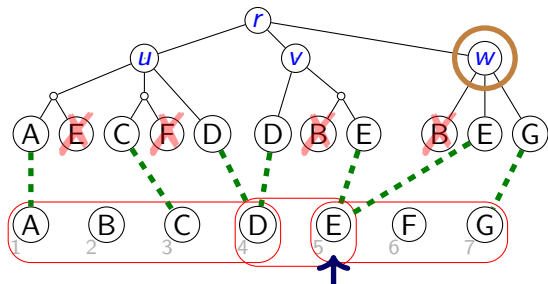
From XP to FPT

- ▶ augment the DP table with **sets of children**,
- ▶ progress **one pivot at a time**

OTDE is FPT for the deletion degree

From XP to FPT

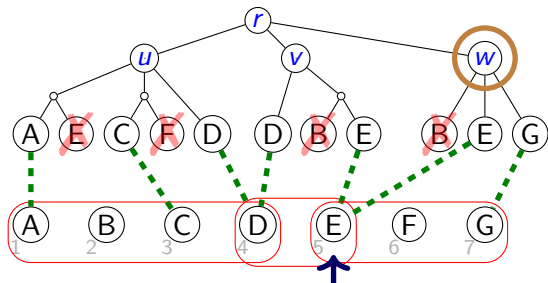
- ▶ augment the DP table with sets of children,
- ▶ progress one pivot at a time



OTDE is FPT for the deletion degree

From XP to FPT

- ▶ augment the DP table with sets of children,
- ▶ progress one pivot at a time

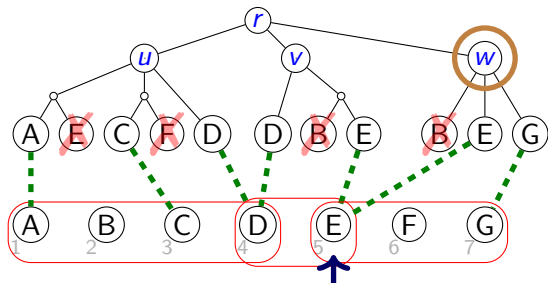


$$\begin{aligned}X(\{r\}, 1, 7) &= X(\{u, v, w\}, 1, 7) \\ &= X(\{u, v\}, 1, 5) + X(\{w\}, 5, 7)\end{aligned}$$

OTDE is FPT for the deletion degree

From XP to FPT

- ▶ augment the DP table with sets of children,
- ▶ progress one pivot at a time



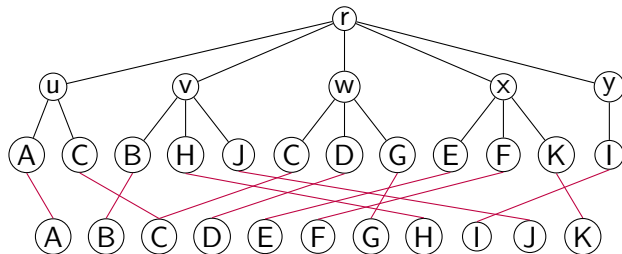
$$\begin{aligned}X(\{r\}, 1, 7) &= X(\{u, v, w\}, 1, 7) \\ &= X(\{u, v\}, 1, 5) + X(\{w\}, 5, 7)\end{aligned}$$

Table size: $2^d n^3$, each entry in $O(dn)$, overall: $O(d2^d n^4)$

OTDE is FPT for the deletion degree

From degree to deletion-degree

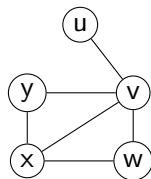
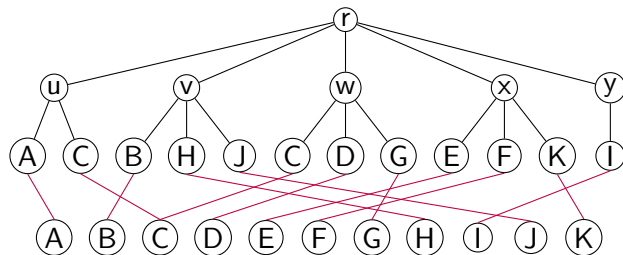
- ▶ only $\partial \ll d$ children with a deletion
- ▶ there exists a large **backbone** without self-conflict



OTDE is FPT for the deletion degree

From degree to deletion-degree

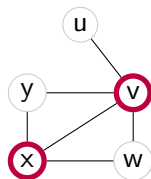
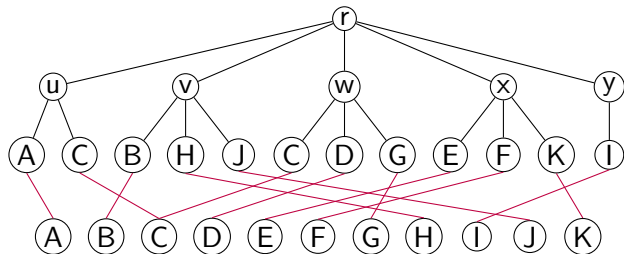
- ▶ only $\partial \ll d$ children with a deletion
- ▶ there exists a large **backbone** without self-conflict
- ▶ compute **some** backbone using Vertex Cover



OTDE is FPT for the deletion degree

From degree to deletion-degree

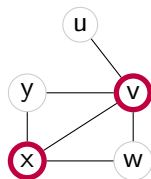
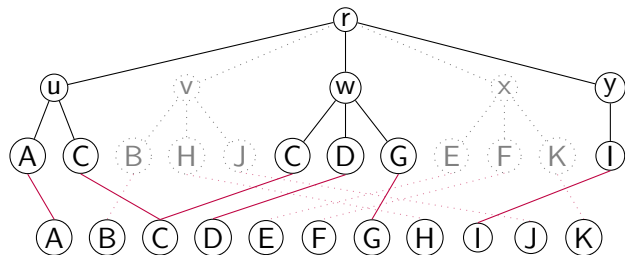
- ▶ only $\partial \ll d$ children with a deletion
- ▶ there exists a large **backbone** without self-conflict
- ▶ compute **some** backbone using Vertex Cover



OTDE is FPT for the deletion degree

From degree to deletion-degree

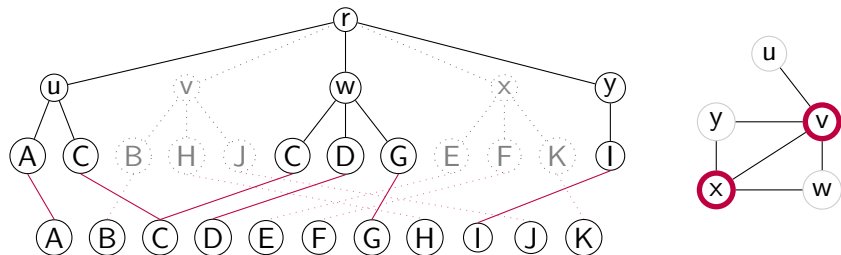
- ▶ only $\partial \ll d$ children with a deletion
- ▶ there exists a large **backbone** without self-conflict
- ▶ compute **some** backbone using Vertex Cover



OTDE is FPT for the deletion degree

From degree to deletion-degree

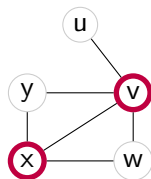
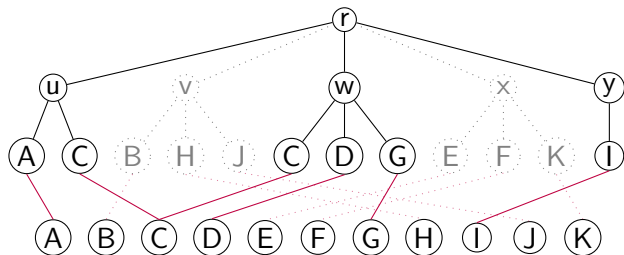
- ▶ only $\partial \ll d$ children with a deletion
- ▶ there exists a large **backbone** without self-conflict
- ▶ compute **some** backbone using Vertex Cover
- ▶ compute DP entries for each
(prefix of the backbone) \cup (any vertices out of the backbone)



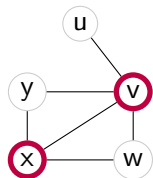
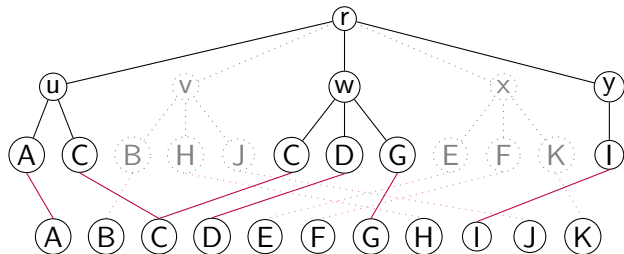
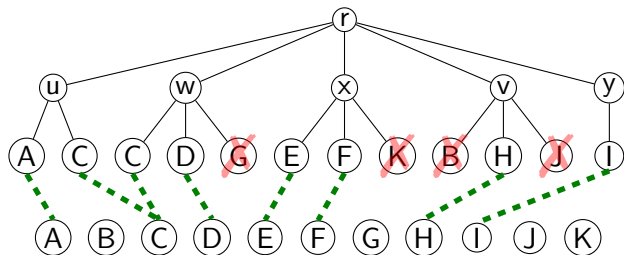
OTDE is FPT for the deletion degree

From degree to deletion-degree

- ▶ only $\partial \ll d$ children with a deletion
- ▶ there exists a large **backbone** without self-conflict
- ▶ compute **some** backbone using Vertex Cover
- ▶ compute DP entries for each
(prefix of the backbone) \cup (any vertices out of the backbone)
- ▶ $2^d \rightarrow d2^\partial$ (+ VC preprocessing in $O(1.3^\partial d + \partial d^2)$)



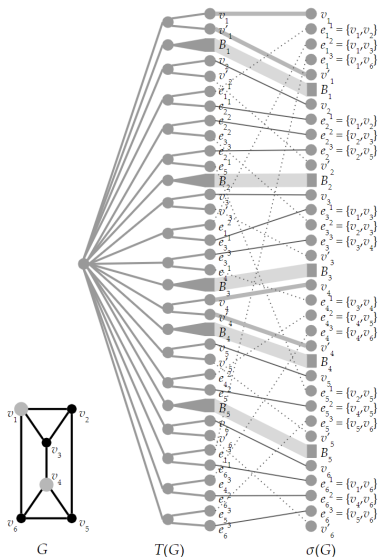
OTDE is FPT for the deletion degree



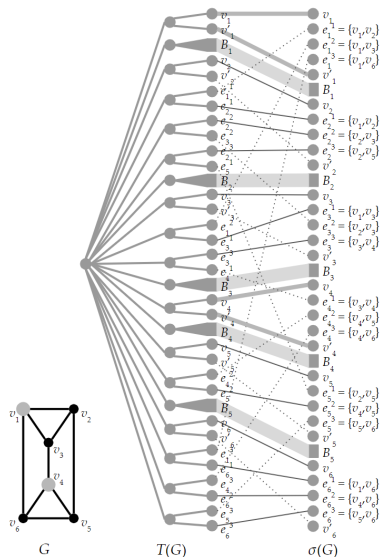
Hardness Results

OTDE is NP-hard: reduction from Independent Set

Given a graph G ,



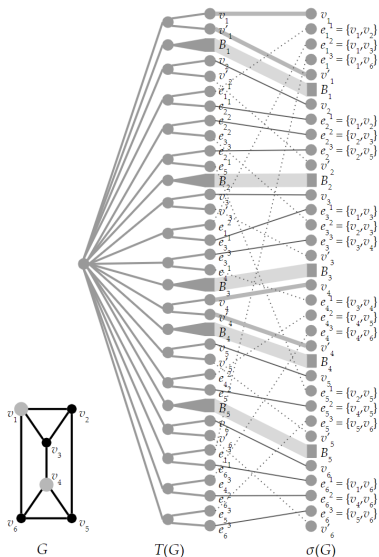
OTDE is NP-hard: reduction from Independent Set



Given a graph G ,
Build tree $T(G)$:

- ▶ One cherry per vertex (u, u')
- ▶ One cherry per edge (e, e')
- ▶ Separators

OTDE is NP-hard: reduction from Independent Set



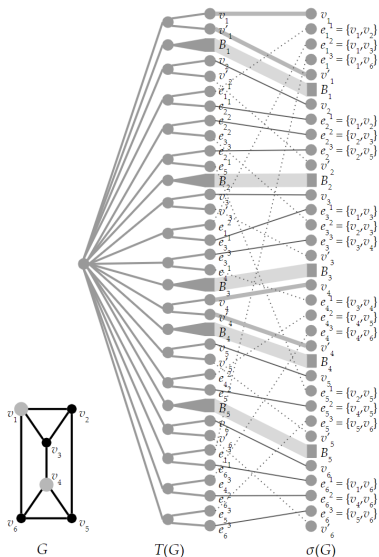
Given a graph G ,
Build tree $T(G)$:

- ▶ One cherry per vertex (u, u')
- ▶ One cherry per edge (e, e')
- ▶ Separators

Build order $\sigma(G)$ (seen as a string):

- ▶ Factor $ue_1e_2e_3u'$ for each vertex and incident edges
- ▶ Separators between factors

OTDE is NP-hard: reduction from Independent Set



Given a graph G ,
Build tree $T(G)$:

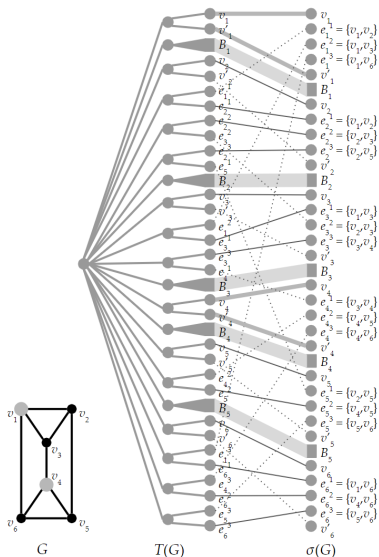
- ▶ One cherry per vertex (u, u')
- ▶ One cherry per edge (e, e')
- ▶ Separators

Build order $\sigma(G)$ (seen as a string):

- ▶ Factor $ue_1e_2e_3u'$ for each vertex and incident edges
- ▶ Separators between factors

Wlog, delete ≤ 1 leaf per cherry,

OTDE is NP-hard: reduction from Independent Set



Given a graph G ,
Build tree $T(G)$:

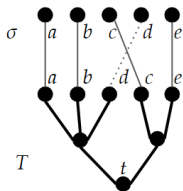
- ▶ One cherry per vertex (u, u')
- ▶ One cherry per edge (e, e')
- ▶ Separators

Build order $\sigma(G)$ (seen as a string):

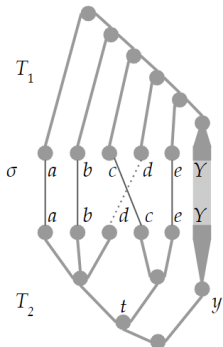
- ▶ Factor $ue_1e_2e_3u'$ for each vertex and incident edges
- ▶ Separators between factors

Wlog, delete ≤ 1 leaf per cherry,
keep both leaves for vertices in
an independent set.

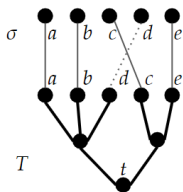
TTDE is NP-hard: reduction from OTDE



Given T, σ



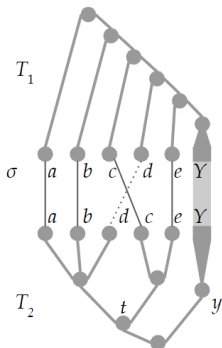
TTDE is NP-hard: reduction from OTDE



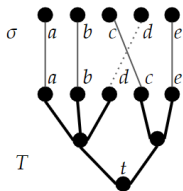
Given T, σ

Build T_1 :

- ▶ Caterpillar following σ
- ▶ Large subtree ("anchor") at the bottom



TTDE is NP-hard: reduction from OTDE



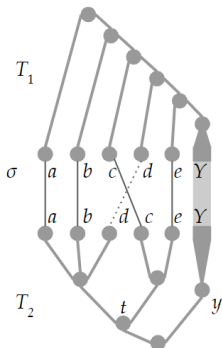
Given T, σ

Build T_1 :

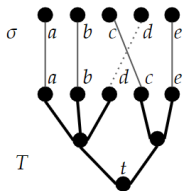
- ▶ Caterpillar following σ
- ▶ Large subtree ("anchor") at the bottom

Build T_2 :

- ▶ Start with T
- ▶ Connect anchor to the root



TTDE is NP-hard: reduction from OTDE



Given T, σ

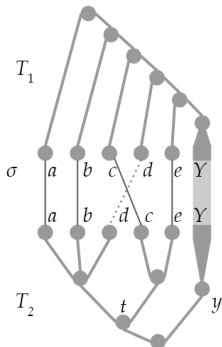
Build T_1 :

- ▶ Caterpillar following σ
- ▶ Large subtree ("anchor") at the bottom

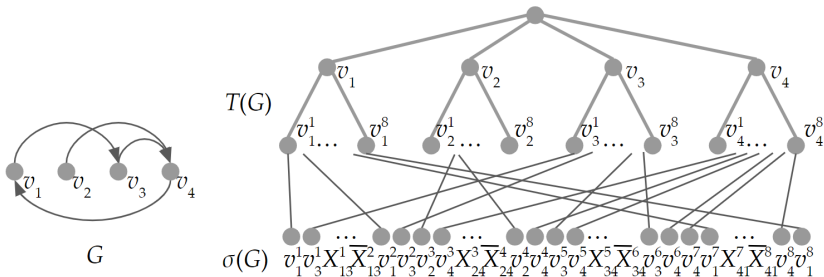
Build T_2 :

- ▶ Start with T
- ▶ Connect anchor to the root

The anchor must be at one end of $T_1 \Rightarrow$ leaf order is the same as σ .

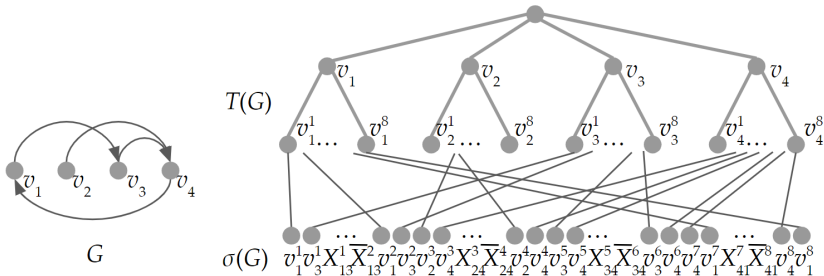


OTCM is NP-hard: reduction from Feedback Arc Set



Given G , build $T(G)$ with one large subtree per vertex.

OTCM is NP-hard: reduction from Feedback Arc Set

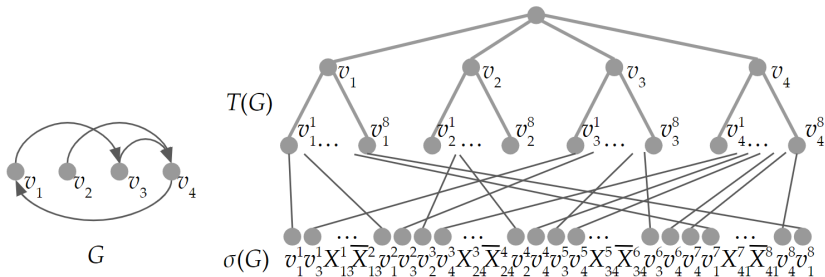


Given G , build $T(G)$ with one large subtree per vertex.

Build $\sigma(G)$ with one factor per arc:

$$v_1 \rightarrow v_3 \implies v_1 v_3 v_2 v_4 v_4 v_2 v_1 v_3$$

OTCM is NP-hard: reduction from Feedback Arc Set



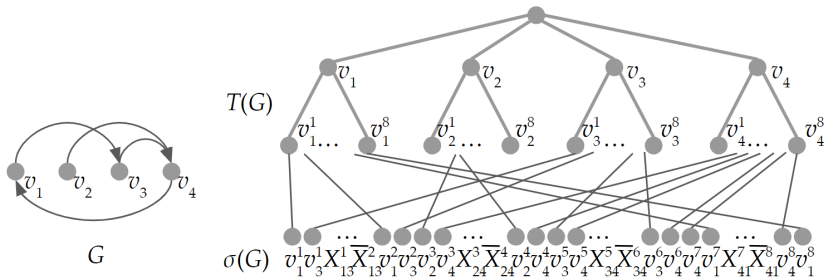
Given G , build $T(G)$ with one large subtree per vertex.

Build $\sigma(G)$ with one factor per arc:

$$v_1 \rightarrow v_3 \implies v_1 v_3 v_2 v_4 v_4 v_2 v_1 v_3$$

Solution: pick a permutation of the vertices

OTCM is NP-hard: reduction from Feedback Arc Set



Given G , build $T(G)$ with one large subtree per vertex.
 Build $\sigma(G)$ with one factor per arc:

$$v_1 \rightarrow v_3 \implies v_1 v_3 v_2 v_4 v_4 v_2 v_1 v_3$$

Solution: pick a permutation of the vertices
 In the arc gadget:

- ▶ v_1, v_3 have 0 crossing if v_1 is before v_3 , 2 otherwise
- ▶ Each other v_i, v_j have 1 crossing.

Experiments

Experiments: data & methods

Data

- ▶ Dated novels of 11 French 19th century writers
- ▶ Distance tables of novels using the relative frequencies of the 500 most frequent tokens
- ▶ Hierarchical clustering based on the distance tables, producing binary trees

Experiments: speed

| tree | # leaves | OTCM time (ms) | # inversions | OTDE time (ms) | # deleted leaves |
|----------|----------|----------------|--------------|----------------|------------------|
| Ségur | 22 | 1 | 40 | 200 | 9 |
| Féval | 23 | 2 | 47 | 268 | 8 |
| Aimard | 24 | 1 | 35 | 401 | 8 |
| Zévaco | 29 | 1 | 42 | 727 | 11 |
| Lesueur | 31 | 1 | 48 | 676 | 13 |
| Zola | 35 | 2 | 60 | 1203 | 9 |
| Gréville | 36 | 2 | 105 | 2211 | 18 |
| Ponson | 42 | 3 | 167 | 3447 | 18 |
| Verne | 58 | 3 | 183 | 13446 | 27 |
| Balzac | 59 | 4 | 248 | 8292 | 34 |
| Sand | 62 | 4 | 283 | 17557 | 39 |

Future work

⇒ Improve the complexity of the dynamic programming algorithm solving OTDE

Experiments: presence of chronological signal

| tree | # leaves | # inversions | ρ_{OTCM} | # deleted leaves | ρ_{OTDE} |
|----------|----------|--------------|---------------|------------------|---------------|
| Séгур | 22 | 40 | 0.24 | 9 | 1 |
| Féval | 23 | 47 | 0.38 | 8 | 0 |
| Aimard | 24 | 35 | 0 | 8 | 0 |
| Zévaco | 29 | 42 | 0 | 11 | 0 |
| Lesueur | 31 | 48 | 0 | 13 | 0 |
| Zola | 35 | 60 | 0 | 9 | 0 |
| Gréville | 36 | 105 | 0 | 18 | 1 |
| Ponson | 42 | 167 | 2.23 | 18 | 0 |
| Verne | 58 | 183 | 0 | 27 | 0 |
| Balzac | 59 | 248 | 0 | 34 | 0 |
| Sand | 62 | 283 | 0 | 39 | 1 |

ρ_{OTCM} (resp. ρ_{OTDE}) = percentage of cases when the best order on the leaves of the tree has the same number of inversions (resp. deleted leaves), or less than the chronological order, among 10000 (resp. 100) randomly generated orders for OTCM (resp. OTDE).

Experiments: identification of noise

Simulation experiment by adding errors in the leaf order

Repeat 100 times:

1. randomly choose “dates” from the interval $[0,999]$
2. build a distance matrix of the absolute differences between “dates” and the corresponding dendrogram
3. insert e artificial errors: pick a new random “date” for e randomly chosen leaves.

► Does OTDE output the set L_e of leaves with artificial errors?

Experiments: identification of noise

| $n =$ # leaves | $e =$ # errors | proportion of cases when $L = L_e$ | when $ L - L_e = 1$ |
|-------------------|-------------------|---------------------------------------|-------------------------|
| 20 | 1 | 0.79 | 1 |
| 20 | 2 | 0.62 | 0.96 |
| 20 | 3 | 0.39 | 0.88 |
| 20 | 4 | 0.33 | 0.77 |
| 20 | 5 | 0.27 | 0.67 |
| 50 | 1 | 0.93 | 1 |
| 50 | 2 | 0.83 | 0.99 |
| 50 | 3 | 0.70 | 0.98 |
| 50 | 4 | 0.59 | 0.91 |
| 50 | 5 | 0.56 | 0.90 |

Observations

- ▶ if at most 2 errors, identified in more than 60% of the experiments, at least 1 identified in more than 96%.

Other Methods to Evaluate the Chronological Signal

What do we want to study/evaluate?

- ▶ the chronological signal in the clustering? (lots of DH tools produce clustering)

What do we want to study/evaluate?

- ▶ the chronological signal in the clustering? (lots of DH tools produce clustering)
- ▶ the chronological signal in the original data? 2 ideas:

What do we want to study/evaluate?

- ▶ the chronological signal in the clustering? (lots of DH tools produce clustering)
- ▶ the chronological signal in the original data? 2 ideas:
 - ▶ how much is the distance matrix Robinsonian?

What do we want to study/evaluate?

- ▶ the chronological signal in the clustering? (lots of DH tools produce clustering)
- ▶ the chronological signal in the original data? 2 ideas:
 - ▶ how much is the distance matrix Robinsonian?
 - ▶ how successful is supervised machine-learning in capturing the chronological signal?

What do we want to study/evaluate?

- ▶ the chronological signal in the clustering? (lots of DH tools produce clustering)
- ▶ the chronological signal in the original data? 2 ideas:
 - ▶ how much is the distance matrix Robinsonian?
 - ▶ how successful is supervised machine-learning in capturing the chronological signal?
- ▶ which linguistic patterns change with the chronology?

How Robinsonian is the input matrix?

Robinsonian matrix

Given a matrix d expressing the distance between novels, we say that d is Robinsonian if for any set of three distinct texts $text_i$, $text_j$ and $text_k$ such that $date(text_i) < date(text_j) < date(text_k)$, $\max(d(text_i, text_j), d(text_j, text_k)) \leq d(text_i, text_k)$.

| | $text_1$ | $text_2$ | $text_3$ |
|----------|----------|----------|----------|
| $text_1$ | 0 | 2 | 4 |
| $text_2$ | 2 | 0 | 1 |
| $text_3$ | 4 | 1 | 0 |

An example of a Robinsonian distance matrix: both $d(text_1, text_2)$ and $d(text_2, text_3)$ are lower than $d(text_1, text_3)$.

How to measure distance?

- ▶ “motifs” : n-grams (unigram to pentagram) of part of speech and semantic labels
- ▶ create vectors of relative frequencies of motifs :
 $p = (p_1, p_2, \dots, p_n), q = (q_1, q_2, \dots, q_n)$
- ▶ canberra metric $D(p, q) = \sum \frac{|p_i - q_i|}{p_i + q_i}$

Example motifs

“Il est fâcheux que cela traîne en longueur”

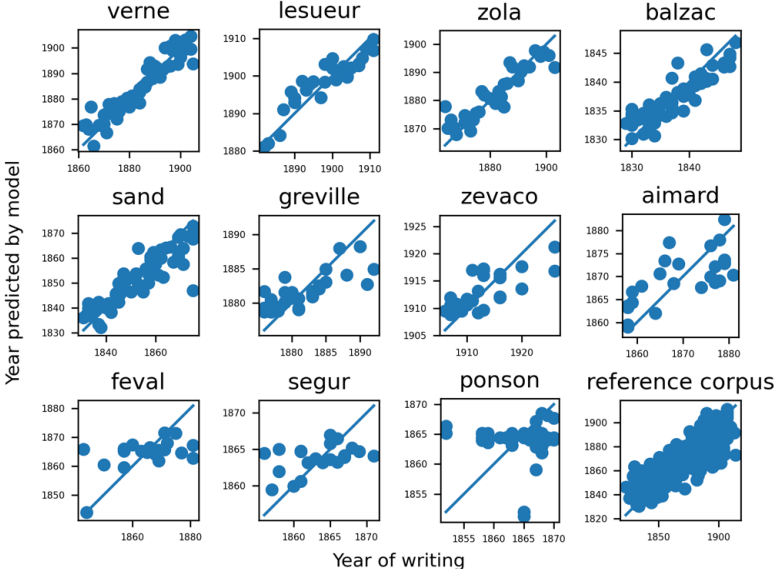
- ▶ Unigrams : [‘il’, ‘être’, ‘ADJ’, ‘que’, ‘cela’, ‘PRES’, ‘en’, ‘NC’, ‘...’]
- ▶ Bigrams : [(‘Il’, ‘être’), (‘être’, ‘ADJ’), (‘ADJ’, ‘que’), (‘que’, ‘cela’), (‘cela’, ‘PRES’), (‘PRES’, ‘en’), (‘en’, ‘NC’), (‘NC’, ‘...’)]

Regression

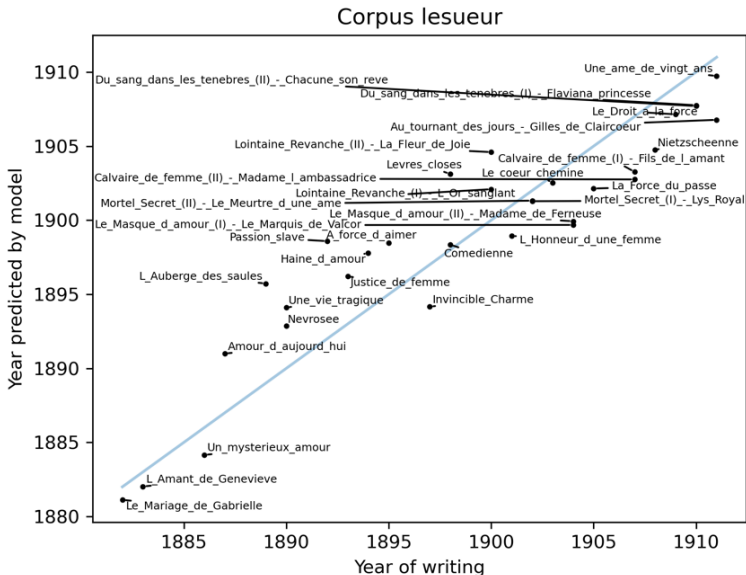
Methodology for the regression

- ▶ Get vector representations of texts with the relative frequency of motifs.
- ▶ Split a corpus of an author in 5 parts: 80 % train, 20 % test. The books are the data-points.
- ▶ Proceed by cross-validation to get predictions on every book.
- ▶ Perform Lasso LARS (regression with feature selection)
- ▶ Study the correlation between the predicted and actual year.
- ▶ Study the remaining features in context and try to interpret them.

Result of the regression



Result of the regression



Which linguistic patterns are increasing or decreasing?

Some patterns are stylemes

- ▶ “. Et” (Zola)
 - ▶ Quoi donc ? Était-ce la fin ? Un souffle glacé avait couru sur le camp, anéanti de sommeil et d'angoisse. **Et** ce fut alors que Jean et Maurice reconnurent le colonel de Vineuil [...] (La débâcle)
 - ▶ What then? Was it the end? An icy breath had run over the camp, annihilated by sleep and anguish. **And** it was then that Jean and Maurice recognized Colonel de Vineuil [...]
- ▶ “dit à [proper_name]” (Balzac)
 - ▶ J'attends la réponse, **dit à Rastignac** le commissionnaire de madame de Nucingen. (Le père Goriot)
 - ▶ I'm waiting for an answer, **said** the commissioner of Madame de Nucingen **to Rastignac**.

Conclusion

Main results

- ▶ NP-hardness proofs for problems useful in bioinformatics and digital humanities
- ▶ FPT-algorithm in the deletion degree
- ▶ implementation in Python of an algorithm solving OTCM and OTDE, to evaluate the chronological signal in a tree
- ▶ a direct method to study the presence of the chronological signal in the data

Conclusion

Future works

- ▶ optimize the dynamic programming algorithm for OTDE
- ▶ evaluate the expected number of inversions or deleted leaves for a random order
- ▶ do more experiments about the new approaches:
 - ▶ solve OTCM / OTDE on other datasets from different fields (some examples already added to https://github.com/oseminck/tree_order_evaluation)
 - ▶ in-depth studies of cases where some leaves are expected to be wrongly ordered for OTDE
 - ▶ discuss the obtained results about the evolution of idiolect with specialists of the authors