Rockafellian Relaxation For Simultaneous Change-Point Detection and Attribution

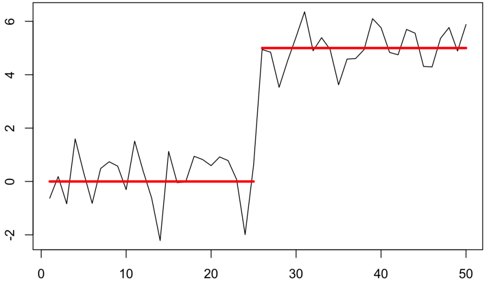
Collaboration with Capital One

Jad Soucar
Dr. Johannes Royset, Dr. Andres Gomez
Department of Industrial and Systems Engineering
University of Southern California

September 16, 2025

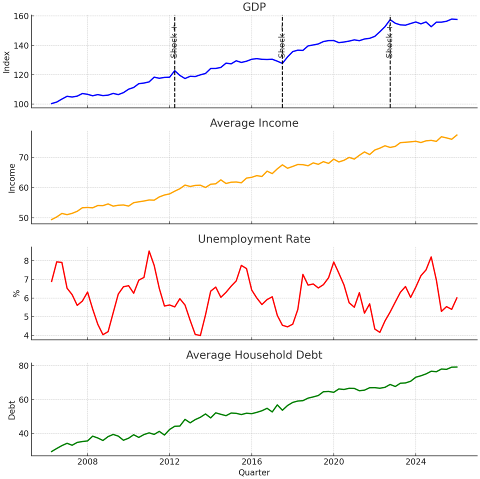
Motivation

- Change-point detection: identifying points in a sequence where there are large structural changes in the statistical properties of the data changes. *Such as mean, volatility, etc.*
- Applications are far reaching and include market regime shifts, volatility changes, fraud detection, policy impact analysis, industrial sensor monitoring, climate regime shifts.



Motivation

- In most cases change point attribution is an equally important question to a practitioner



Problem

Given

- A *target* time series $z = \{z^t \in \mathbb{R}\}_{t=1,\dots,T}$
- d *feature* timeseries $X = (x_{(1)}, \dots x_{(d)}), \quad \{x_{(i)}^t \in \mathbb{R}\}_{t=1,\dots,T}$

Task

Identify change points in z while also attributing each change point to a subset $S \subseteq \{1, \dots, d\}$ of feature timeseries

Previous Work

- Approaches to **change point detection** include kernel, probabilistic, subspace, machine learning classifier, and graph based models[1]

[1] Samaneh Aminikhanghahi and Diane J. Cook (Sept. 2016). "A survey of methods for time series change point detection". In: *Knowledge and Information Systems* 51.2, pp. 339–367. ISSN: 0219-3116. DOI: 10.1007/s10115-016-0987-z. URL: http://dx.doi.org/10.1007/s10115-016-0987-z.

[2] Shimeng Huang, Jonas Peters, and Niklas Pfister (2024). *Causal Change Point Detection and Localization*. DOI: 10.48550/ARXIV.2403.12677. URL: https://arxiv.org/abs/2403.12677.

Previous Work

- Approaches to **change point detection** include kernel, probabilistic, subspace, machine learning classifier, and graph based models[1]
- Fastest and most popular approaches include Pelt, Segment Neighborhood, and Binary segmentation.

$$\{0 = \tau_0 < \tau_1 < \cdots < \tau_r = T\} = \underset{r, \tau_0 < \cdots < \tau_r}{\operatorname{argmin}} \sum_{i=1}^{r} C(\tau_{i-1}, \tau_i) + r\gamma$$

Notice that the number of change points r is specified by the user!

[1] Samaneh Aminikhanghahi and Diane J. Cook (Sept. 2016). "A survey of methods for time series change point detection". In: *Knowledge and Information Systems* 51.2, pp. 339–367. ISSN: 0219-3116. DOI: 10.1007/s10115-016-0987-z. URL: http://dx.doi.org/10.1007/s10115-016-0987-z.

[2] Shimeng Huang, Jonas Peters, and Niklas Pfister (2024). *Causal Change Point Detection and Localization*. DOI: 10.48550/ARXIV.2403.12677. URL: https://arxiv.org/abs/2403.12677.

Previous Work

- Approaches to **change point detection** include kernel, probabilistic, subspace, machine learning classifier, and graph based models[1]
- Fastest and most popular approaches include Pelt, Segment Neighborhood, and Binary segmentation.

$$\{0 = \tau_0 < \tau_1 < \cdots < \tau_r = T\} = \underset{r, \tau_0 < \cdots < \tau_r}{\operatorname{argmin}} \sum_{i=1}^{r} C(\tau_{i-1}, \tau_i) + r\gamma$$

Notice that the number of change points r is specified by the user!

Obstacle to Simultaneous Change Point Detection and Attribution

Popular approaches limit the structure of C from being coupled across segments!

- **Change Point Attribution** involves post hoc statistical tests on each segment[2].

[1] Samaneh Aminikhanghahi and Diane J. Cook (Sept. 2016). "A survey of methods for time series change point detection". In: *Knowledge and Information Systems* 51.2, pp. 339–367. ISSN: 0219-3116. DOI: 10.1007/s10115-016-0987-z. URL: http://dx.doi.org/10.1007/s10115-016-0987-z.

[2] Shimeng Huang, Jonas Peters, and Niklas Pfister (2024). *Causal Change Point Detection and Localization*. DOI: 10.48550/ARXIV.2403.12677. URL: https://arxiv.org/abs/2403.12677.

Loss Function

Definition (Curve Fitting)

Given

- $X, \beta \in \mathbb{R}^{d \times T}, z \in \mathbb{R}^T$
- $\ell(z, w) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$
- $\phi(x, \beta) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$
- $\theta, \lambda, \gamma \geq 0$

Then for all $k \leq T$

$$\mathcal{L}_k(\beta) = \sum_{t=1}^{k} \ell\big(z^t, \phi(X_t, \beta^t)\big) + \theta \sum_{t=1}^{k} \|\beta^t\|_* + \lambda \sum_{t=2}^{k} \|\beta^t - \beta^{t-1}\|_\star$$

Loss Function

Definition (Rockafellian Perturbation)

Given

- $X, \beta \in \mathbb{R}^{d \times T}$, $u \in \mathbb{R}^{d \times T-1}$, $z \in \mathbb{R}^T$
- $\ell(z, w) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$
- $\phi(x, \beta) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$
- $\theta, \lambda, \gamma \geq 0$

Then for all $k \leq T$

$$\mathcal{L}_k(\beta, u) = \sum_{t=1}^{k} \ell\left(z^t, \phi(X_t, \beta^t)\right) + \theta \sum_{t=1}^{k} \|\beta^t\|_* + \lambda \sum_{t=2}^{k} \|\beta^t - \beta^{t-1} + u^t\|_\star$$

Loss Function

Definition (Rockafellian Perturbation)

Given

- $X, \beta \in \mathbb{R}^{d \times T}, u \in \mathbb{R}^{d \times T-1}, z \in \mathbb{R}^T$
- $\ell(z, w) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$
- $\phi(x, \beta) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$
- $\theta, \lambda, \gamma \geq 0$

Then for all $k \leq T$

$$\mathcal{L}_k(\beta, u) = \sum_{t=1}^{k} \ell\left(z^t, \phi(X_t, \beta^t)\right) + \theta \sum_{t=1}^{k} \|\beta^t\|_* + \lambda \sum_{t=2}^{k} \|\beta^t - \beta^{t-1} + u^t\|_\star + \gamma \sum_{t=2}^{k} \mathbb{I}_{\{u^t \neq 0\}}$$

Loss Function

Definition (Rockafellian Perturbation)

Given

- $X, \beta \in \mathbb{R}^{d \times T}, u \in \mathbb{R}^{d \times T-1}, z \in \mathbb{R}^{T}$
- $\ell(z, w) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$
- $\phi(x, \beta) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$
- $\theta, \lambda, \gamma \geq 0$

Then for all $k \leq T$

$$\mathcal{L}_k(\beta, u) = \sum_{t=1}^{k} \ell(z^t, \phi(X_t, \beta^t)) + \theta \sum_{t=1}^{k} \|\beta^t\|_* + \lambda \sum_{t=2}^{k} \|\beta^t - \beta^{t-1} + u^t\|_\star + \gamma \sum_{t=2}^{k} \mathbb{I}_{\{u^t \neq 0\}}$$

Attribution Score

$$a_i = u_i^{*\tau} / \|u^{*\tau}\|_1, \quad \forall i \in \{1, \ldots, d\} \text{ Where } \tau \text{ is change point of } \{z^t\}_{t=1}^{T}$$

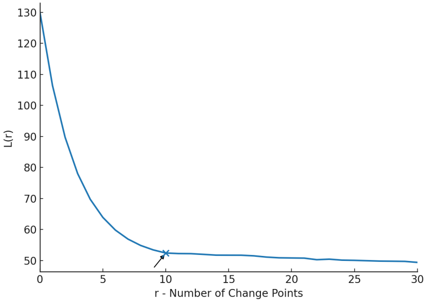
Advantages

- **Advantages**
  - Our objective allows for segment coupling $\rightarrow$ We can performs simultaneous change point detection and attribution
  - The user does not specify the number of change points

Advantages

- **Advantages**
  - Our objective allows for segment coupling → We can performs simultaneous change point detection and attribution
  - The user does not specify the number of change points

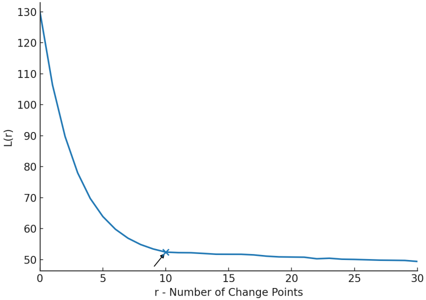
$$L(r) = \min_{r,\tau_0<\cdots<\tau_r} \sum_{i=1}^{r} C(\tau_{i-1}, \tau_i) + r\gamma$$



Advantages

- **Advantages**
  - Our objective allows for segment coupling $\rightarrow$ We can performs simultaneous change point detection and attribution
  - The user does not specify the number of change points $\rightarrow$ One-Shot identification of the optimal number of change points

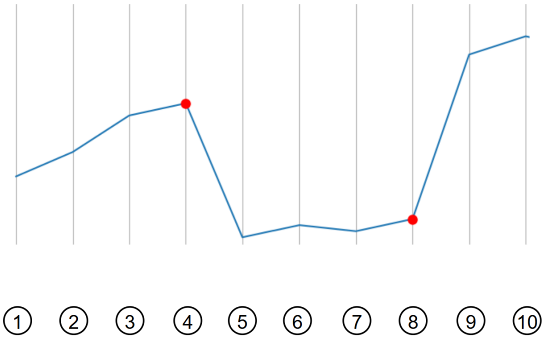
$$L(r) = \min_{r, \tau_0 < \cdots < \tau_r} \sum_{i=1}^{r} C(\tau_{i-1}, \tau_i) + r\gamma$$



Solution - In Theory

Definition (Loss Function)

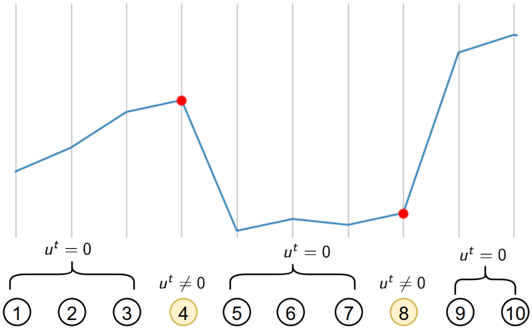
$$\mathcal{L}_k(\beta, u) = \sum_{t=1}^{k} \ell(z^t, \phi(X_t, \beta^t)) + \theta \sum_{t=1}^{k} \|\beta^t\|_* + \lambda \sum_{t=2}^{k} \|\beta^t - \beta^{t-1} + u^t\|_\star + \gamma \sum_{t=2}^{k} \mathbb{I}_{\{u^t \neq 0\}}$$



Solution - In Theory

Definition (Loss Function)

$$\mathcal{L}_k(\beta, u) = \sum_{t=1}^{k} \ell(z^t, \phi(X_t, \beta^t)) + \theta \sum_{t=1}^{k} \|\beta^t\|_* + \lambda \sum_{t=2}^{k} \|\beta^t - \beta^{t-1} + u^t\|_\star + \gamma \sum_{t=2}^{k} \mathbb{I}_{\{u^t \neq 0\}}$$



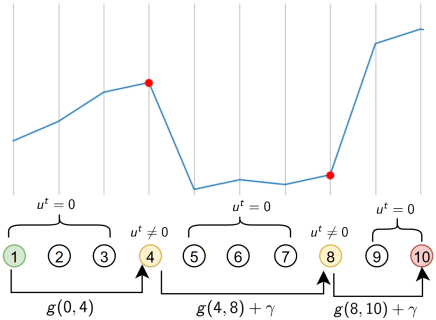
Solution - In Theory

Definition (Loss Function)

$$\mathcal{L}_k(\beta, u) = \sum_{t=1}^{k} \ell(z^t, \phi(X_t, \beta^t)) + \theta \sum_{t=1}^{k} \|\beta^t\|_* + \lambda \sum_{t=2}^{k} \|\beta^t - \beta^{t-1} + u^t\|_\star + \gamma \sum_{t=2}^{k} \mathbb{I}_{\{u^t \neq 0\}}$$

Definition (Partial Loss Function)

$$g(i,j) = \inf_\beta \sum_{t=i+1}^{j} \ell(z^t, \phi(X_t, \beta^t)) + \theta \sum_{t=i+1}^{j} \|\beta^t\|_* + \lambda \sum_{t=i+2}^{j} \|\beta^t - \beta^{t-1}\|_\star, \quad i < j$$



Problem - Reformulations

Original Problem

minimize$_{\beta,u}\, \mathcal{L}_k(\beta, u) =$
$\sum_{t=1}^{k} \ell\big(z^t, \phi(X_t, \beta^t)\big) + \theta \sum_{t=1}^{k} \|\beta^t\|_* + \lambda \sum_{t=2}^{k} \|\beta^t - \beta^{t-1} + u^t\|_\star + \gamma \sum_{t=2}^{k} \mathbb{I}_{\{u^t \neq 0\}}$

Problem - Reformulations

Original Problem

$\text{minimize}_{\beta,u}\, \mathcal{L}_k(\beta, u) =$
$\sum_{t=1}^{k} \ell(z^t, \phi(X_t, \beta^t)) + \theta \sum_{t=1}^{k} \|\beta^t\|_* + \lambda \sum_{t=2}^{k} \|\beta^t - \beta^{t-1} + u^t\|_\star + \gamma \sum_{t=2}^{k} \mathbb{I}_{\{u^t \neq 0\}}$

Shortest Path Problem

Find Shortest path on a graph $G = (V, E)$. Where $V = \{0, \dots, k\}$, $E = \{(i, j) : i, j \in V, j > i\}$, and weights $w_{ij} = g(i, j) + \gamma \mathbb{I}_{i \neq 0}$

Problem - Reformulations

Original Problem

minimize$_{\beta,u}\, \mathcal{L}_k(\beta, u) =$
$\sum_{t=1}^{k} \ell(z^t, \phi(X_t, \beta^t)) + \theta \sum_{t=1}^{k} \|\beta^t\|_* + \lambda \sum_{t=2}^{k} \|\beta^t - \beta^{t-1} + u^t\|_\star + \gamma \sum_{t=2}^{k} \mathbb{I}_{\{u^t \neq 0\}}$

Shortest Path Problem

Find Shortest path on a graph $G = (V, E)$. Where $V = \{0, \ldots, k\}$, $E = \{(i, j) : i, j \in V, j > i\}$, and weights $w_{ij} = g(i, j) + \gamma \mathbb{I}_{i \neq 0}$

Theorem (Dynamic Programming Reformulation)

Given $\mathcal{L}_k$ *and* $g(i, j)$, *then* $\min_{\beta,u} \mathcal{L}_j(\beta, u) = f(j)$

$$f(j) = \min \left\{ g(0, j), \min_{1 \leq i \leq j-1} f(i) + g(i, j) + \gamma \right\}, \quad j \leq k$$

Loss Function - Reformulation

Shortest Path Formulation

The problem of minimizing $\mathcal{L}_k$ reduces to finding the shortest path on a graph $G = (V, E)$. Where $V = \{0, \ldots, k\}$, $E = \{(i,j) : i,j \in V, j > i\}$, and weights $w_{ij} = g(i,j) + \gamma \mathbb{I}_{i \neq 0}$

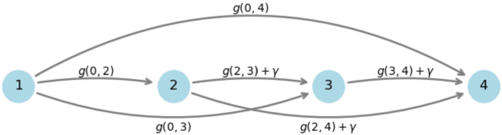


Figure: Example where $T = 4$

- Candidate algorithms include Dijkstra, Bellman-Ford, A*

Bottleneck

Bottleneck

Computing $g(i,j)$ is an expensive task if ϕ and ℓ are non-convex. For a timeseries of length T, one would need $T(T+1)/2$ evaluations of g

Definition (Partial Loss Function)

$$g(i,j) = \inf_\beta \sum_{t=i+1}^{j} \ell(z^t, \phi(X_t, \beta^t)) + \theta \sum_{t=i+1}^{j} \|\beta^t\|_* + \lambda \sum_{t=i+2}^{j} \|\beta^t - \beta^{t-1}\|_\star, \quad i < j$$

Bottleneck

Bottleneck

Computing $g(i,j)$ is an expensive task if ϕ and ℓ are non-convex. For a timeseries of length T, one would need $T(T+1)/2$ evaluations of g

Definition (Partial Loss Function)

$g(i,j) = \inf_\beta \sum_{t=i+1}^{j} \ell(z^t, \phi(X_t, \beta^t)) + \theta \sum_{t=i+1}^{j} \|\beta^t\|_* + \lambda \sum_{t=i+2}^{j} \|\beta^t - \beta^{t-1}\|_\star, \quad i < j$

Theorem (Linearization of the temporal regularization term in g)

Given $g(i,j)$, we have the following lower bound where y is an arbitrary vector in $\mathbb{B}_\star = \{y \in \mathbb{R}^{j-i-1} : \|y\|_\star \leq 1\}$

$$g(i,j) \geq g_{LB}(i,j;y) = \inf_\beta \sum_{t=i+1}^{j} \ell(z^t, \phi(X_t, \beta^t)) + \theta \sum_{t=i+1}^{j} \|\beta^t\|_* + \lambda \sum_{t=i+2}^{j} \langle y^t, \beta^t - \beta^{t-1} \rangle$$

Solving the Relaxed $g(i, j)$

- There are several primal-dual algorithms capable of solving for β, u and y for $g_{lb}(i, j; y)$.
  - Chambolle-Pock[3]
  - Proximal Alternating Direction Method of Multiplier (ADMM)[4]
  - Proximal Alternating Linearized Minimization (iPALM)[5]
- Most primal-dual algorithms require a user-specified number of iterations

$$O(z, X, n) \to g_{LB}(i, j; y) \leq g(i, j)$$

[3] Sebastian Banert, Manu Upadhyaya, and Pontus Giselsson (2023). *The Chambolle–Pock method converges weakly with gt*; $1/2$ *and* $||L||^2 lt; 4/(1+2)$. DOI: 10.48550/ARXIV.2309.03998. URL: https://arxiv.org/abs/2309.03998.

[4] Yu Yang et al. (2022). "Proximal ADMM for nonconvex and nonsmooth optimization". In: *Automatica* 146, p. 110551. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2022.110551. URL: https://www.sciencedirect.com/science/article/pii/S0005109822004125.

[5] Thomas Pock and Shoham Sabach (2017). "Inertial Proximal Alternating Linearized Minimization (iPALM) for Nonconvex and Nonsmooth Problems". In: DOI: 10.48550/ARXIV.1702.02505. URL: https://arxiv.org/abs/1702.02505.

Quick Review

1. **Goal**: Change Point Detection and Attribution
2. Introduced an optimization problem

$$\mathcal{L}_k(\beta, u) = \sum_{t=1}^{k} \ell(z^t, \phi(X_t, \beta^t)) + \theta \sum_{t=1}^{k} \|\beta^t\|_* + \lambda \sum_{t=2}^{k} \|\beta^t - \beta^{t-1} + u^t\|_\star + \gamma \sum_{t=2}^{k} \mathbb{I}_{\{u^t \neq 0\}}$$

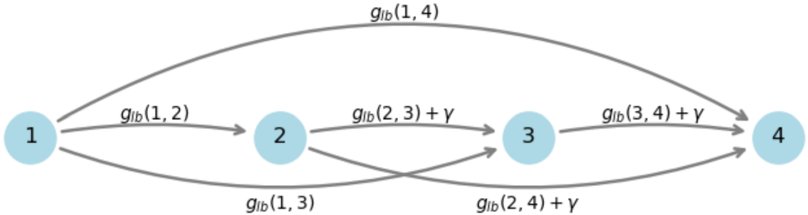
3. Equivalent to solving a shortest path problem $G = (V, E)$. Where $V = \{1, \ldots, k\}$, $E = \{(i,j) : i, j \in V, j > i\}$, and weights

$$w_{ij} = g(i,j) + \gamma \mathbb{I}_{i \neq 0}$$

4. Estimate edge weights with lower bounds that are simpler to compute

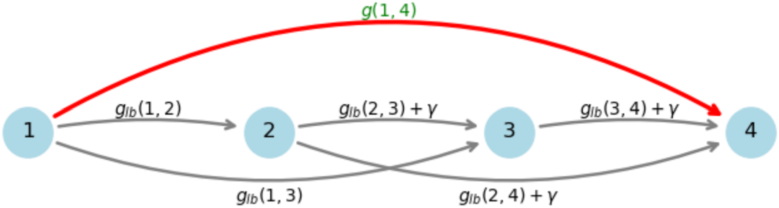
$$g(i,j) \geq g_{LB}(i,j;y) = \inf_{\beta} \sum_{t=i+1}^{j} \ell(z^t, \phi(X_t, \beta^t)) + \theta \sum_{t=i+1}^{j} \|\beta^t\|_* + \lambda \sum_{t=i+2}^{j} \langle y^t, \beta^t - \beta^{t-1} \rangle$$

Proposed Algorithm - Simple Pruning with Lower Bounds



Proposed Algorithm - Simple Pruning with Lower Bounds

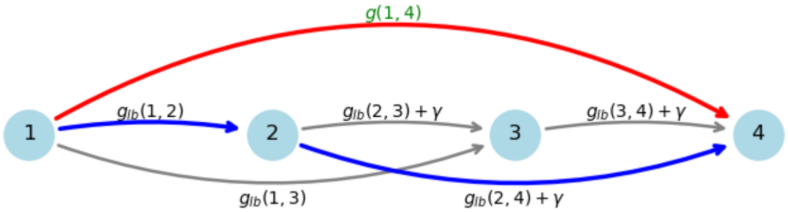
- Let $\pi = (1 \rightarrow 4)$ be our starting path.
- Compute $g(i, j)$'s included in our path $\rightarrow$ update graph and current cost



- $C(\pi) = g(1, 4)$

Proposed Algorithm - Simple Pruning with Lower Bounds

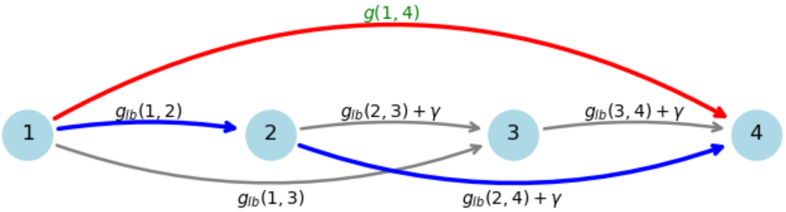
- Let $\pi = (1 \rightarrow 4)$ be our starting pat, Next "insert" arc path $(1 \rightarrow 2)$



- $g(1, 4) > g_{lb}(1, 2) + g_{lb}(2, 4) + \gamma$

Proposed Algorithm - Simple Pruning with Lower Bounds

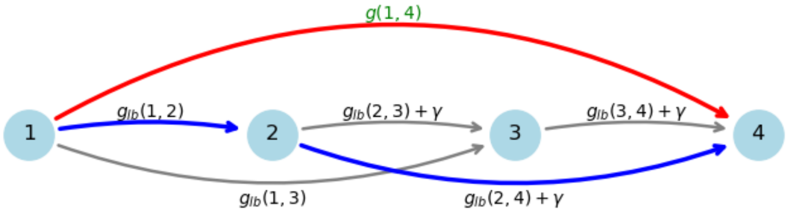
- Let $\pi = (1 \rightarrow 4)$ be our starting pat, Next "insert" arc path $(1 \rightarrow 2)$



- $g(1,4) > g_{lb}(1,2) + g_{lb}(2,4) + \gamma$
  - **If True:** $g(1,4) > g(1,2) + g(2,4) + \gamma$
    - **If True:** Update graph, $\pi \leftarrow (1,2,4)$, $C(\pi) \leftarrow g(1,2) + g(2,4) + \gamma$

Proposed Algorithm - Simple Pruning with Lower Bounds

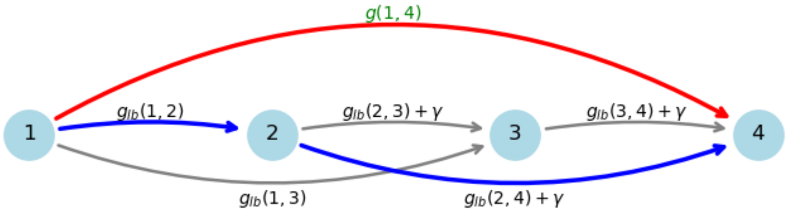
- Let $\pi = (1 \rightarrow 4)$ be our starting pat, Next “insert” arc path $(1 \rightarrow 2)$



- $g(1,4) > g_{lb}(1,2) + g_{lb}(2,4) + \gamma$
  - **If True:** $g(1,4) > g(1,2) + g(2,4) + \gamma$
    - **If True:** Update graph, $\pi \leftarrow (1,2,4)$, $C(\pi) \leftarrow g(1,2) + g(2,4) + \gamma$
    - **If False:** Update graph Wasted Evaluation of $g(1,2)$!

Proposed Algorithm - Simple Pruning with Lower Bounds

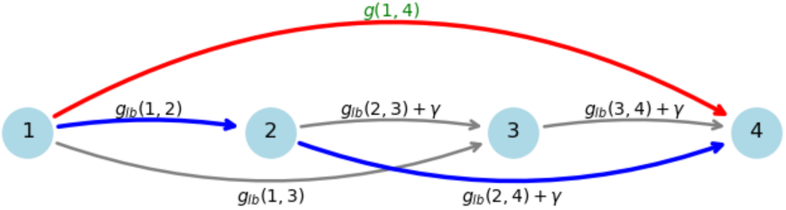
- Let $\pi = (1 \rightarrow 4)$ be our starting pat, Next "insert" arc path $(1 \rightarrow 2)$



- $g(1,4) > g_{lb}(1,2) + g_{lb}(2,4) + \gamma$
  - **If True:** $g(1,4) > g(1,2) + g(2,4) + \gamma$
    - **If True:** Update graph, $\pi \leftarrow (1,2,4)$, $C(\pi) \leftarrow g(1,2) + g(2,4) + \gamma$
    - **If False:** Update graph Wasted Evaluation of $g(1,2)$!
  - **If False:** We know that $g(1,4) < g(1,2) + g(2,4) + \gamma$ so arc $(1,2)$ will be pruned.

Proposed Algorithm - Simple Pruning with Lower Bounds

- Let $\pi = (1 \rightarrow 4)$ be our starting pat, Next "insert" arc path $(1 \rightarrow 2)$



- $g(1,4) > g_{lb}(1,2) + g_{lb}(2,4) + \gamma$
  - **If True:** $g(1,4) > g(1,2) + g(2,4) + \gamma$
    - **If True:** Update graph, $\pi \leftarrow (1,2,4)$, $C(\pi) \leftarrow g(1,2) + g(2,4) + \gamma$
    - **If False:** Update graph Wasted Evaluation of $g(1,2)$!
  - **If False:** We know that $g(1,4) < g(1,2) + g(2,4) + \gamma$ so arc $(1,2)$ will be pruned.
- Repeat this process for all arcs in topological order *(Next arc would be* $1 \rightarrow 3$*)*

Simple Pruning with Lower Bound

- **Data.** Set $j = 1$, Some feasible path $\pi = (0 = i_0 < i_1 < \cdots < i_r = k)$
- **Setup.** Set $G_{ij} \leftarrow g_{lb}(i, j)$ for all $0 \leq i < j \leq k$. Set $F_0 = 0$. For each $1 \leq j \leq k$ set F_j, where $s(j) = \max\{0 \leq q \leq r : i_q \leq j\}$

$$F_j \leftarrow [g(s(j), j) + \mathbb{I}_{\{i_{s(j)} \neq 0\}}] + \sum_{q=1}^{s(j)} \left(g(i_{q-1}, i_q) + \gamma \cdot \mathbb{I}_{\{i_{q-1} \neq 0\}} \right)$$

- **Step 1.** For $i = 0, \ldots, j - 1$ check if $F_j > F_i + G_{i,j} + \gamma \mathbb{I}_{i \neq 0}$. If true set $G_{ij} \leftarrow g(i, j)$ and update F_j as

$$F_j \leftarrow \min\{F_j, F_i + G_{i,j} + \gamma \mathbb{I}_{i \neq 0}\}$$

- **Step 2.** If $j = k$ **terminate**. Otherwise update $j \leftarrow j + 1$ and return to step 1

Numerical Results

For the purposes of our numerical experiments we make the following assumptions

- $z, \beta \in \mathbb{R}^T$, $u \in \mathbb{R}^{T-1}$
- $\mathcal{L}_k(\beta, u) = \sum_{t=1}^{k} |z^t - \beta^t| + \theta \sum_{t=1}^{k} \|\beta^t\|_1 + \lambda \sum_{t=2}^{k} \|\beta^t - \beta^{t-1} + u^t\|_1 + \gamma \sum_{t=2}^{k} \mathbb{I}_{\{u^t \neq 0\}}$
- We use Chambolle-Pock to calculate $g_{lb}(i, j)$

Numerical Results

For the purposes of our numerical experiments we make the following assumptions

- $z, \beta \in \mathbb{R}^T, u \in \mathbb{R}^{T-1}$
- $\mathcal{L}_k(\beta, u) = \sum_{t=1}^k |z^t - \beta^t| + \theta \sum_{t=1}^k \|\beta^t\|_1 + \lambda \sum_{t=2}^k \|\beta^t - \beta^{t-1} + u^t\|_1 + \gamma \sum_{t=2}^k \mathbb{I}_{\{u^t \neq 0\}}$
- We use Chambolle-Pock to calculate $g_{lb}(i, j)$

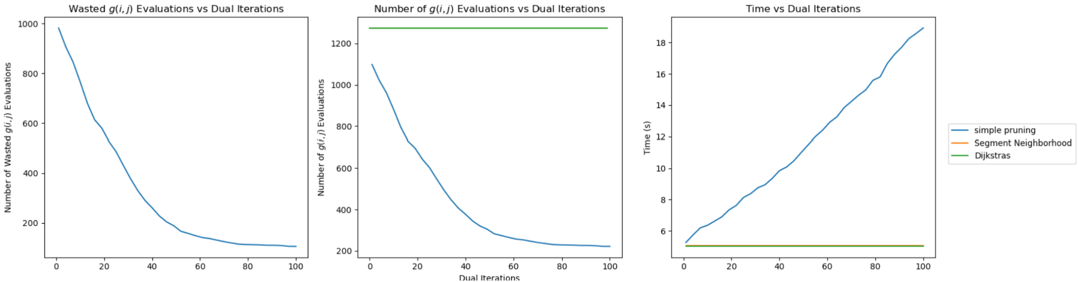


Figure: Set $\lambda = .8, \theta = .1, \gamma = 3$. We average the number of g evaluations and the number of "wasted" g evaluations across 100 samples of synthetic data. We also test Segment Neighborhood, and Dijkstra algorithms without lower bounds.

Future Work

- Test algorithm with non-linear/non-convex/non-smooth ϕ
  - Determine whether our algorithm is faster then traditional solvers (MIP, Non-Lazy Path Finding)
  - Determine whether the number of g evaluations still decreases with the number of dual iterations

Future Work

- Test algorithm with non-linear/non-convex/non-smooth ϕ
  - Determine whether our algorithm is faster then traditional solvers (MIP, Non-Lazy Path Finding)
  - Determine whether the number of g evaluations still decreases with the number of dual iterations
- Test algorithm on $\mathcal{L}_k$ that incorporates X to observe the attribution scoring mechanism

Future Work

- Test algorithm with non-linear/non-convex/non-smooth ϕ
  - Determine whether our algorithm is faster then traditional solvers (MIP, Non-Lazy Path Finding)
  - Determine whether the number of g evaluations still decreases with the number of dual iterations
- Test algorithm on $\mathcal{L}_k$ that incorporates X to observe the attribution scoring mechanism
- Find conditions on X and z that guarantees low g evaluation and wasted evaluation rates
- Explainable formulation of the objective function for industry practitioners

Questions!