Flow-based Attribution in Graphical Models: A Recursive Shapley Approach

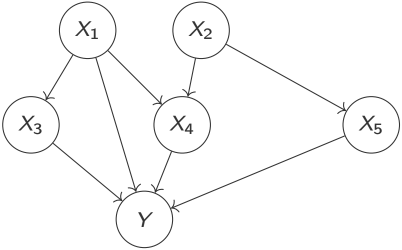
Raghav Singal

Amazon

Joint work with George Michailidis and Hoiyi Ng

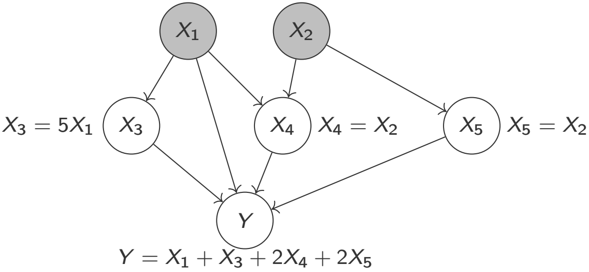
ICML 2021

Motivating Example



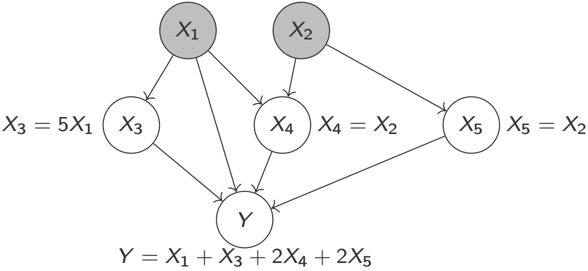
Directed acyclic graph (DAG)

Motivating Example



Structural equations

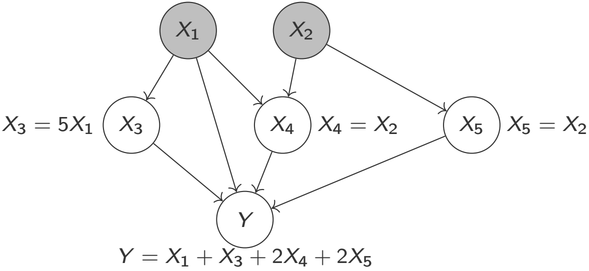
Motivating Example



Structural equations

- suppose *source variables* (X_1, X_2) change from $(0, 0)$ to $(1, 1)$
- as a result, output Y changes from 0 to 10, i.e., *effect* equals 10

Motivating Example

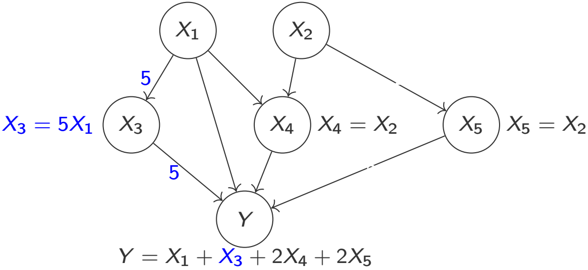


Structural equations

- suppose *source variables* (X_1, X_2) change from $(0, 0)$ to $(1, 1)$
- as a result, output Y changes from 0 to 10, i.e., *effect* equals 10

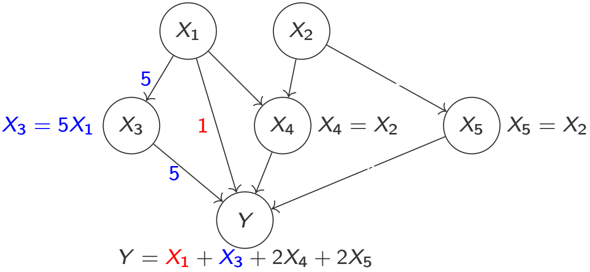
How does the effect (change in Y) flow through the graph?

Motivating Example



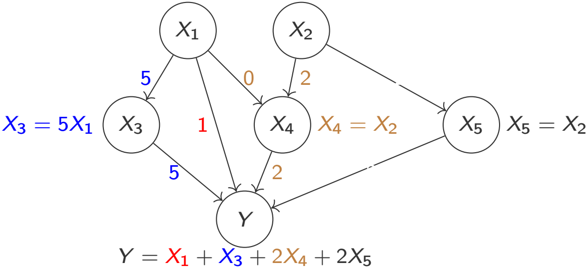
Quantifying effect propagation for a *linear model*

Motivating Example



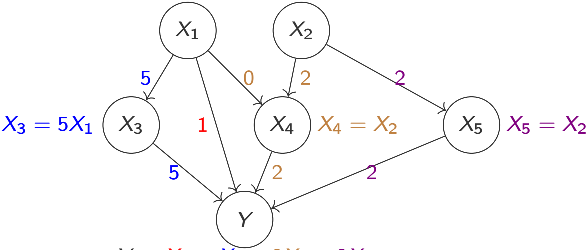
Quantifying effect propagation for a *linear model*

Motivating Example



Quantifying effect propagation for a *linear model*

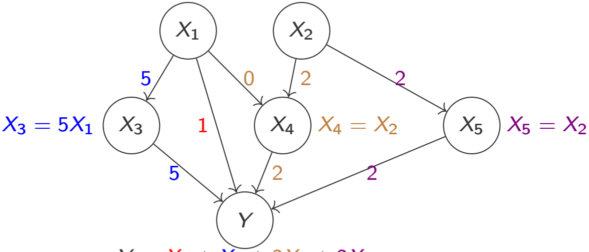
Motivating Example



$$Y = X_1 + X_3 + 2X_4 + 2X_5$$

Quantifying effect propagation for a *linear model*

Motivating Example

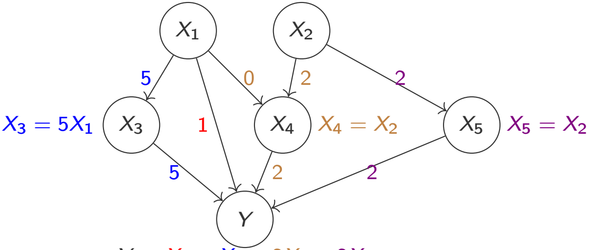


$$Y = X_1 + X_3 + 2X_4 + 2X_5$$

Quantifying effect propagation for a *linear model*

But what if the structural equations are *non-linear*?
Can we develop a model-agnostic flow-based attribution method?

Motivating Example



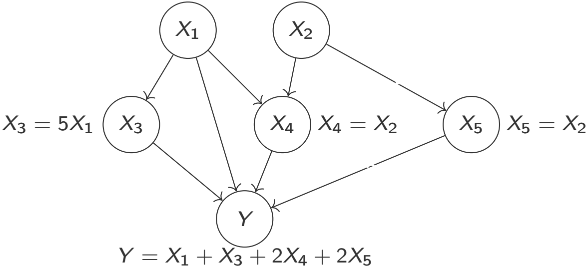
$$Y = X_1 + X_3 + 2X_4 + 2X_5$$

Quantifying effect propagation for a *linear model*

But what if the structural equations are *non-linear*?
Can we develop a model-agnostic flow-based attribution method?

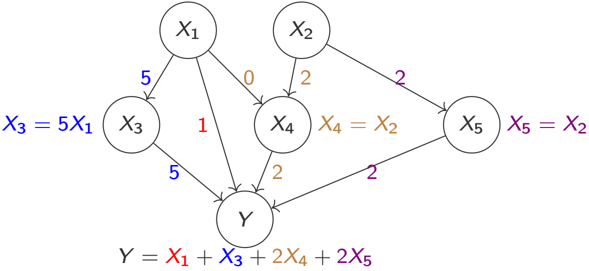
Applications: (1) interpretable ML (neural nets) and (2) causality (mediation)

Flow-based Axioms



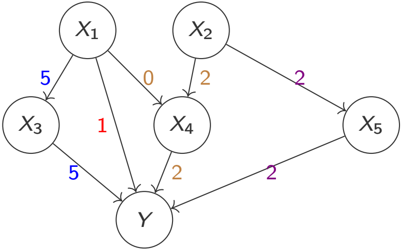
- consider the linear model from before for ease of illustration
- recall (X_1, X_2) changes from $(0, 0)$ to $(1, 1)$
- as a result, Y changes from 0 to 10

Flow-based Axioms



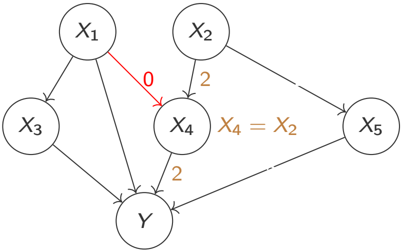
Recall the "natural" flow for a linear model

Flow-based Axioms



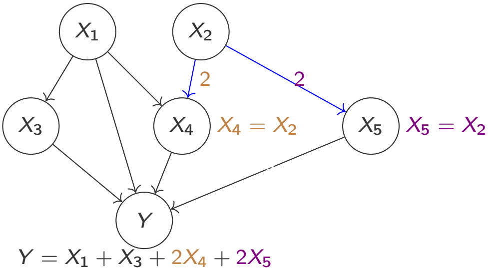
- **flow conservation**: at each node, flow in equals flow out [Bach et al., 2015]

Flow-based Axioms



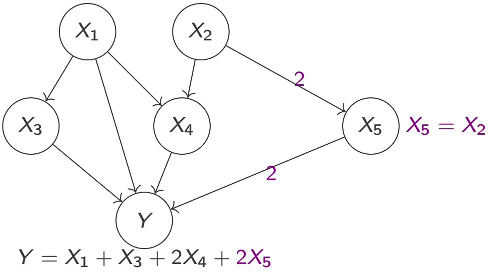
- **flow conservation**: at each node, flow in equals flow out
- **flow nullity**: "redundant" edge receives zero flow

Flow-based Axioms



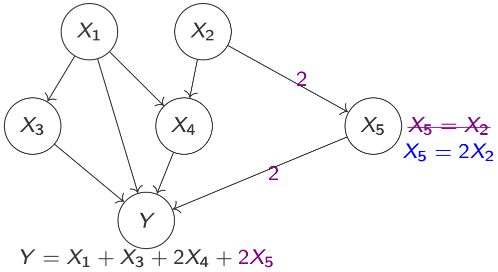
- **flow conservation**: at each node, flow in equals flow out
- **flow nullity**: "redundant" edge receives zero flow
- **flow symmetry**: "equivalent" edges receive the same flow

Flow-based Axioms



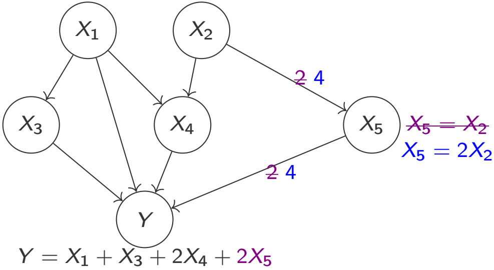
- **flow conservation**: at each node, flow in equals flow out
- **flow nullity**: "redundant" edge receives zero flow
- **flow symmetry**: "equivalent" edges receive the same flow
- **flow linearity**: attribution is robust to "linear pertubations"

Flow-based Axioms



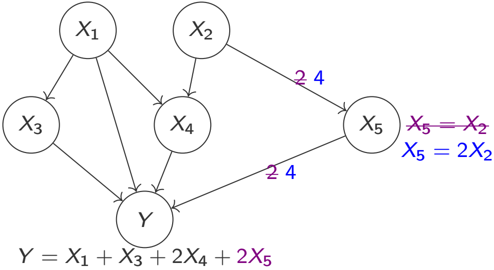
- **flow conservation**: at each node, flow in equals flow out
- **flow nullity**: "redundant" edge receives zero flow
- **flow symmetry**: "equivalent" edges receive the same flow
- **flow linearity**: attribution is robust to "linear pertubations"

Flow-based Axioms



- **flow conservation**: at each node, flow in equals flow out
- **flow nullity**: "redundant" edge receives zero flow
- **flow symmetry**: "equivalent" edges receive the same flow
- **flow linearity**: attribution is robust to "linear pertubations"

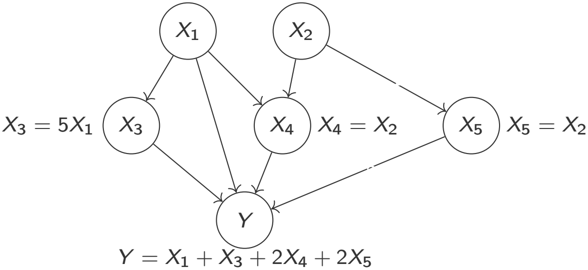
Flow-based Axioms



- **flow conservation**: at each node, flow in equals flow out
- **flow nullity**: "redundant" edge receives zero flow
- **flow symmetry**: "equivalent" edges receive the same flow
- **flow linearity**: attribution is robust to "linear pertubations"

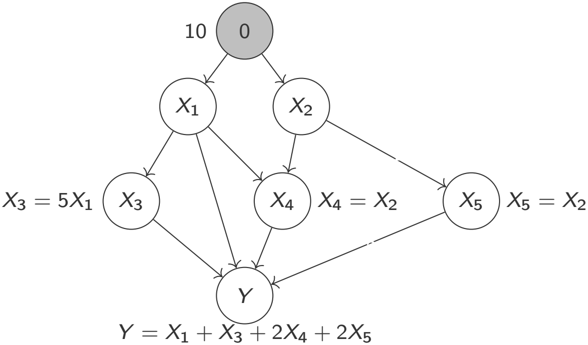
Punchline: there exists a *unique* solution to these four axioms

Our Approach: Recursive Shapley Value (RSV)



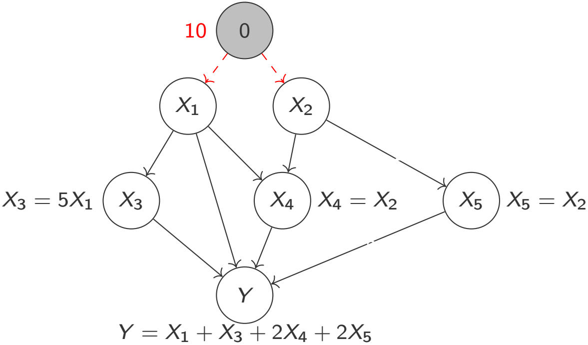
- same running example
- recall (X_1, X_2) changes from $(0, 0)$ to $(1, 1)$
- as a result, Y changes from 0 to 10 (i.e., effect equals 10)

Our Approach: Recursive Shapley Value (RSV)



- insert a dummy node 0 and "attribute" all the effect (10) to it

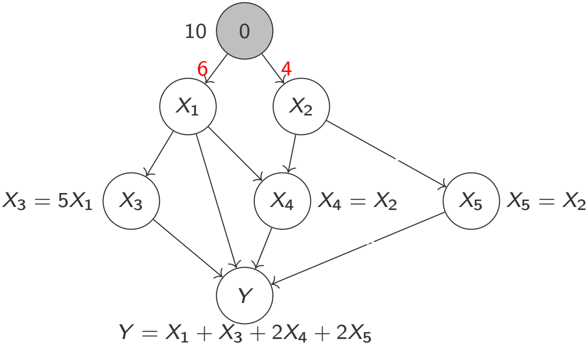
Our Approach: Recursive Shapley Value (RSV)



- insert a dummy node 0 and "attribute" all the effect (10) to it
- **step 0**: evaluate "contributions" of outgoing edges of node 0 via Shapley value

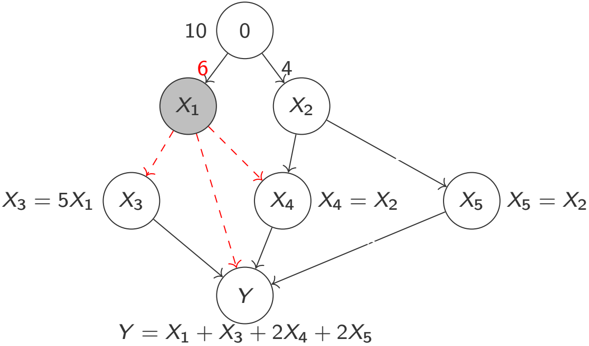
What would have been the effect had edge $(0, 1)$ not propagated the change?

Our Approach: Recursive Shapley Value (RSV)



- insert a dummy node 0 and "attribute" all the effect (10) to it
- **step 0**: evaluate "contributions" of outgoing edges of node 0 via Shapley value

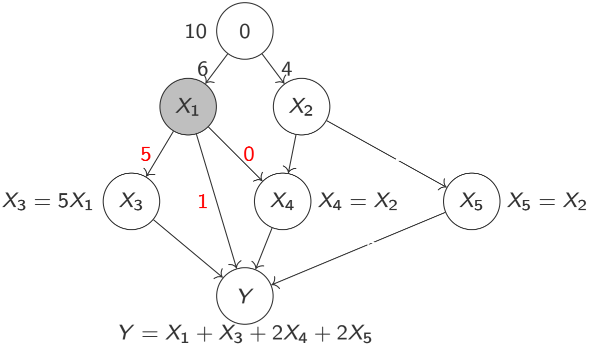
Our Approach: Recursive Shapley Value (RSV)



- insert a dummy node 0 and "attribute" all the effect (10) to it
- **step 0**: evaluate "contributions" of outgoing edges of node 0 via Shapley value
- **step 1**: evaluate "contributions" of outgoing edges of node 1 via Shapley value

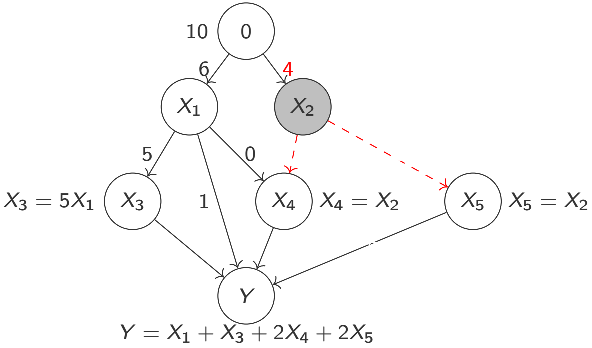
How much attribution would node 1 have received if edge $(1, 3)$
had not propagated the change at node 1? (recursive!)

Our Approach: Recursive Shapley Value (RSV)



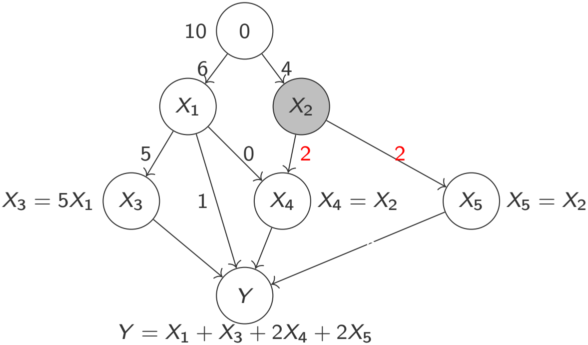
- insert a dummy node 0 and "attribute" all the effect (10) to it
- **step 0**: evaluate "contributions" of outgoing edges of node 0 via Shapley value
- **step 1**: evaluate "contributions" of outgoing edges of node 1 via Shapley value

Our Approach: Recursive Shapley Value (RSV)



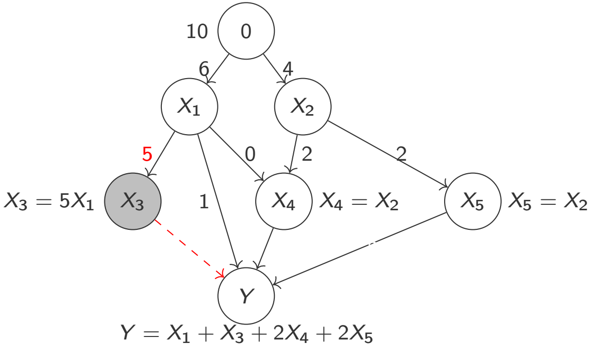
- insert a dummy node 0 and "attribute" all the effect (10) to it
- **step 0**: evaluate "contributions" of outgoing edges of node 0 via Shapley value
- **step 1**: evaluate "contributions" of outgoing edges of node 1 via Shapley value
- **step 2**

Our Approach: Recursive Shapley Value (RSV)



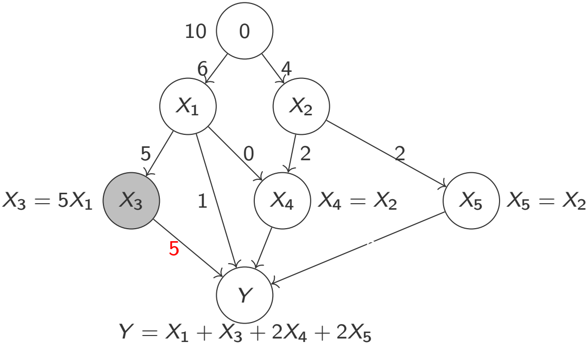
- insert a dummy node 0 and "attribute" all the effect (10) to it
- **step 0**: evaluate "contributions" of outgoing edges of node 0 via Shapley value
- **step 1**: evaluate "contributions" of outgoing edges of node 1 via Shapley value
- **step 2**

Our Approach: Recursive Shapley Value (RSV)



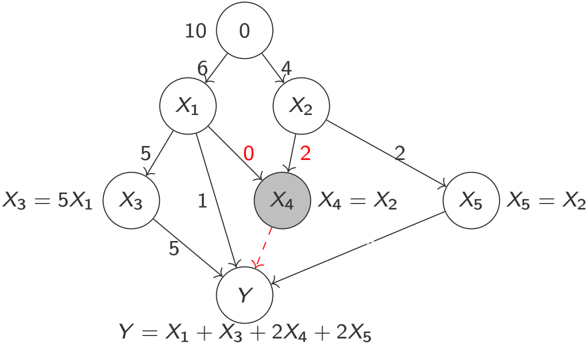
- insert a dummy node 0 and "attribute" all the effect (10) to it
- **step 0**: evaluate "contributions" of outgoing edges of node 0 via Shapley value
- **step 1**: evaluate "contributions" of outgoing edges of node 1 via Shapley value
- **step 2**
- **step 3**

Our Approach: Recursive Shapley Value (RSV)



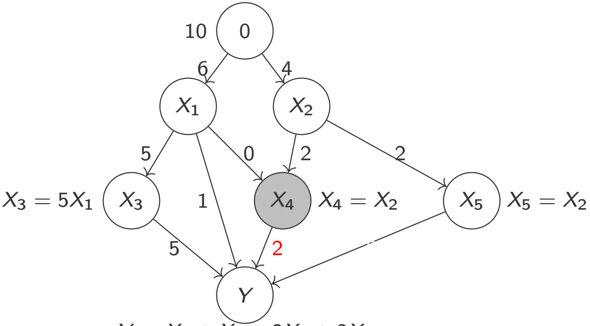
- insert a dummy node 0 and "attribute" all the effect (10) to it
- **step 0**: evaluate "contributions" of outgoing edges of node 0 via Shapley value
- **step 1**: evaluate "contributions" of outgoing edges of node 1 via Shapley value
- **step 2**
- **step 3**

Our Approach: Recursive Shapley Value (RSV)



- insert a dummy node 0 and "attribute" all the effect (10) to it
- **step 0**: evaluate "contributions" of outgoing edges of node 0 via Shapley value
- **step 1**: evaluate "contributions" of outgoing edges of node 1 via Shapley value
- **step 2**
- **step 3**
- **step 4**

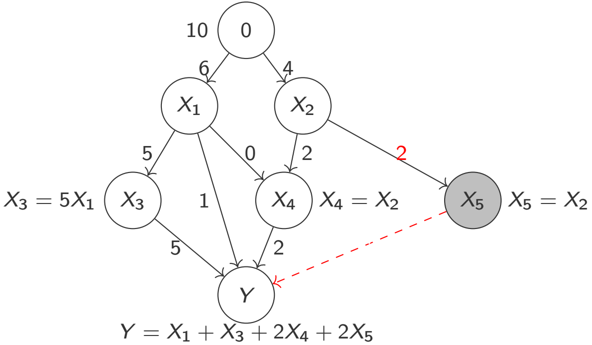
Our Approach: Recursive Shapley Value (RSV)



$$Y = X_1 + X_3 + 2X_4 + 2X_5$$

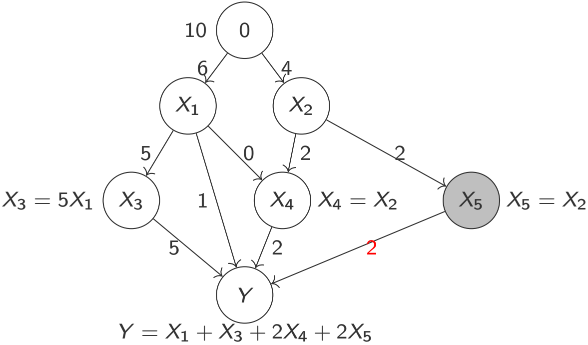
- insert a dummy node 0 and "attribute" all the effect (10) to it
- **step 0**: evaluate "contributions" of outgoing edges of node 0 via Shapley value
- **step 1**: evaluate "contributions" of outgoing edges of node 1 via Shapley value
- **step 2**
- **step 3**
- **step 4**

Our Approach: Recursive Shapley Value (RSV)



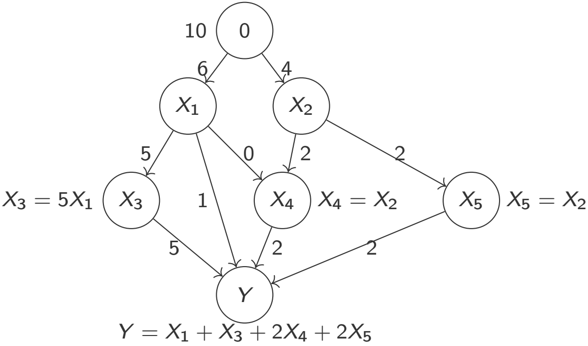
- insert a dummy node 0 and "attribute" all the effect (10) to it
- **step 0**: evaluate "contributions" of outgoing edges of node 0 via Shapley value
- **step 1**: evaluate "contributions" of outgoing edges of node 1 via Shapley value
- **step 2**
- **step 3**
- **step 4**
- **step 5**

Our Approach: Recursive Shapley Value (RSV)



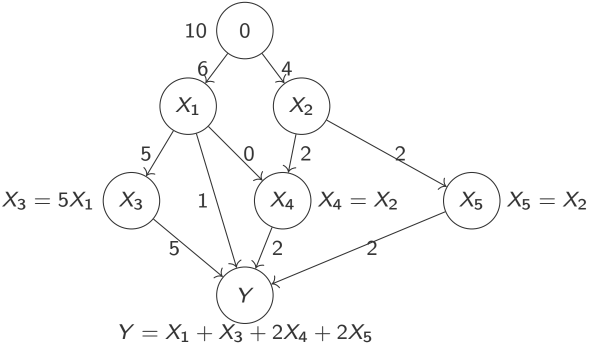
- insert a dummy node 0 and "attribute" all the effect (10) to it
- **step 0**: evaluate "contributions" of outgoing edges of node 0 via Shapley value
- **step 1**: evaluate "contributions" of outgoing edges of node 1 via Shapley value
- **step 2**
- **step 3**
- **step 4**
- **step 5**

Our Approach: Recursive Shapley Value (RSV)



- our "top-down" philosophy is fundamentally different from "bottom-up"
- sanity check: RSV recovers the "natural" flow for a linear model
- in fact, it is the *only* solution to the flow-based axioms

Our Approach: Recursive Shapley Value (RSV)



- our "top-down" philosophy is fundamentally different from "bottom-up"
- sanity check: RSV recovers the "natural" flow for a linear model
- in fact, it is the *only* solution to the flow-based axioms

> **Theorem: Axioms**
>
> RSV is the unique solution to the four flow-based axioms

Additional Properties

Implementation invariance: robustness to internal changes in the graph
[Sundararajan et al., 2017]

Sensitivity: if output (in)dependent on an input, then so should be attribution
[Sundararajan et al., 2017]

Monotonicity: if output monotone in an input, then so should be attribution
[Sundararajan & Najmi, 2020]

Affine scale invariance: robustness to input scalings (Celsius vs. Fahrenheit)
[Sundararajan & Najmi, 2020]

Additional Properties

Implementation invariance: robustness to internal changes in the graph
[Sundararajan et al., 2017]

Sensitivity: if output (in)dependent on an input, then so should be attribution
[Sundararajan et al., 2017]

Monotonicity: if output monotone in an input, then so should be attribution
[Sundararajan & Najmi, 2020]

Affine scale invariance: robustness to input scalings (Celsius vs. Fahrenheit)
[Sundararajan & Najmi, 2020]

Proposition: Properties

RSV obeys implementation invariance, sensitivty, monotonicity, and ASI

Additional Properties

Implementation invariance: robustness to internal changes in the graph
[Sundararajan et al., 2017]

Sensitivity: if output (in)dependent on an input, then so should be attribution
[Sundararajan et al., 2017]

Monotonicity: if output monotone in an input, then so should be attribution
[Sundararajan & Najmi, 2020]

Affine scale invariance: robustness to input scalings (Celsius vs. Fahrenheit)
[Sundararajan & Najmi, 2020]

Proposition: Properties
RSV obeys implementation invariance, sensitivty, monotonicity, and ASI

In addition, generalizes a number of existing node-based approaches

Concluding Remarks

Summary

- formalized the attribution problem over a graphical model
- highlighted limitations of existing methods
- developed a model-agnostic flow-based attribution method (RSV)
- uniquely satisfies a set of flow-based axioms + four desirable properties
- recovers existing approaches for the "natural" use cases
- facilitates mediation analysis in non-linear models

Concluding Remarks

Summary

- formalized the attribution problem over a graphical model
- highlighted limitations of existing methods
- developed a model-agnostic flow-based attribution method (RSV)
- uniquely satisfies a set of flow-based axioms + four desirable properties
- recovers existing approaches for the "natural" use cases
- facilitates mediation analysis in non-linear models

Ongoing research

- extending the framework to a probabilistic graph [Pearl, 2009]
- connections to the causality literature [Pearl, 2001; Chockler & Halpern, 2005]
- computational tractability (beyond linear models)

Concluding Remarks

Summary

- formalized the attribution problem over a graphical model
- highlighted limitations of existing methods
- developed a model-agnostic flow-based attribution method (RSV)
- uniquely satisfies a set of flow-based axioms + four desirable properties
- recovers existing approaches for the "natural" use cases
- facilitates mediation analysis in non-linear models

Ongoing research

- extending the framework to a probabilistic graph [Pearl, 2009]
- connections to the causality literature [Pearl, 2001; Chockler & Halpern, 2005]
- computational tractability (beyond linear models)

Thank you! rs3566@columbia.edu

https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3845526

References

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek.
On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation.
PLOS ONE, 10(7):e0130140, 2015.

Hana Chockler and Joseph Y Halpern.
Responsibility and Blame: A Structural-Model Approach.
Journal of Artificial Intelligence Research, 22:93–115, 2004.

Judea Pearl.
Direct and Indirect Effects.
Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence, 32:411–420, 2001.

Judea Pearl.
Causality.
Cambridge University Press, 2009.

Mukund Sundararajan and Amir Najmi.
The Many Shapley Values for Model Explanation.
In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 9269–9278. PMLR, 2020.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan.
Axiomatic Attribution for Deep Networks.
In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 3319–3328. PMLR, 2017.