NETWORK INTRUSION DETECTION USING MACHINE LEARNING METHODS

Grant Parker

Gregg Puttkammer

Jimson Huang

Nathan Yao

PURDUE UNIVERSITY | Department of Computer Science

Watch out Purdue!

From: Kupka, Alexandra Lynn <kupka@purdue.edu>
Sent: Thursday, August 3, 2023 6:33 PM
Subject: PART TIME/ FULL TIME JOB OFFER

You are invited to participate in a Part-time work and study offer for current staff or students. It is an opportunity to earn up to $350 weekly. The job is exciting, flexible and will not affect your current job or studies. Don't miss this opportunity, For more details copy and paste the link below

COPY AND PASTE THE LINK BELOW TO YOUR ADDRESS WEB BAR

docs.google.com/forms/d/e/1FAIpQLSeBCSiuU0795CeD86lwRb5jPn0t0V73IfFGlVN1xO-x3wVgDg/viewform

Grant Parker

Introduction and Background

Why Intrusion Detection Systems (IDS) Matters

[3]

- Cyber Security Risks
  - 2,365 cyber attacks in 2023 with 343,338,964 victims[1]
  - Average cost of data breach is $4.45 million[1]
  - McAfee estimates losses of over $1 trillion per year[2]
- Intrusion detection systems work on a system level
  - Can be used to detect malware, anomalous network traffic, and system outliers
  - Can also provide insight into system vulnerabilities and highlight areas to watch

Cyber espionage is “the greatest transfer of wealth in history” – NSA Director Kieth Alexander

Dataset

UNSW-NB15 Dataset

- Combination of real network traffic and synthetic attacks
- 43 explanatory variables
  - Time, service, packet count, ttl, duration
- 2 response variables
  - Binary (malicious/benign)
  - Categorical (if malicious, what type of attack)
    - Wide range of attack behaviors
- 257,675 observations
  - Pre-split into 175,342 observations for training
  - 82,333 observations for testing

UNSW SYDNEY

ACCS Cyber Range Lab

Current State of Methods

Study of Intrusion Detection Methods

- Tendency towards older datasets
  - 56% KDDcup99, NSL-KDD datasets
  - Different attacks detected: Exploits, DoS, Probe, Worms
- Great variety of feature extraction methods and classifiers used
  - 60% use feature extraction
    - PCA, Auto-Encoder, Naïve Bayes
  - Popular classifiers:
    - Convolutional Neural Network (CNN)
    - Deep Belief Network (DBN)
    - Deep Neural Network (DNN)
    - ML: Random Forests, SVM, Ensemble
      - Usually paired with DL algorithms

[4] [5]

Current State of Methods Cont.

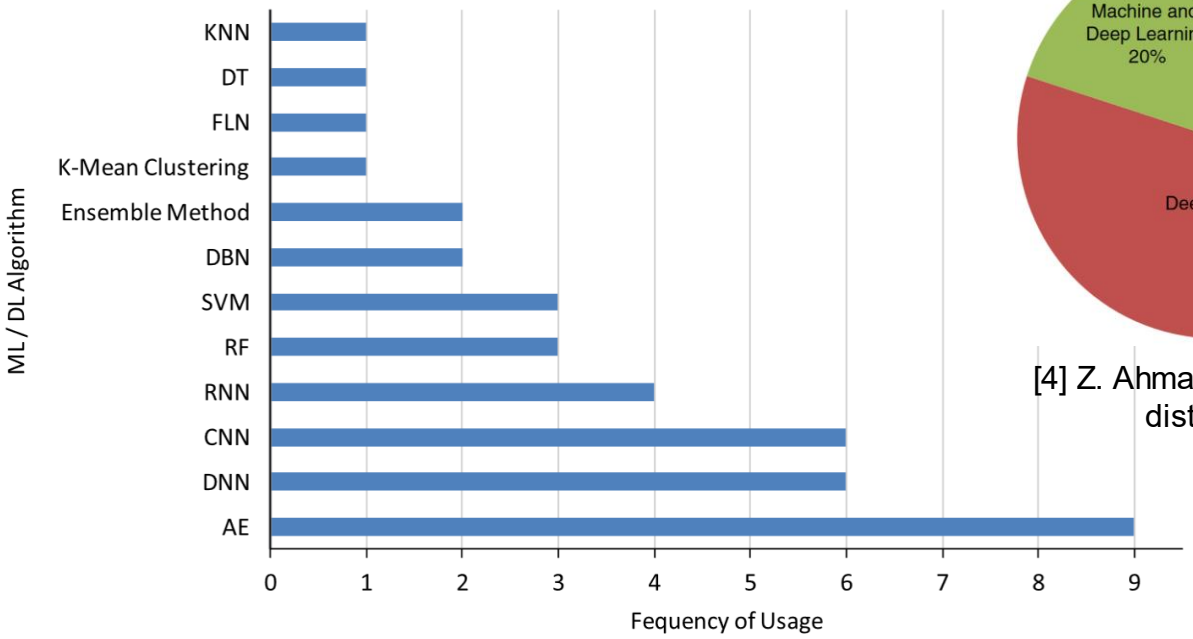
Study of Intrusion Detection Methods: Observations

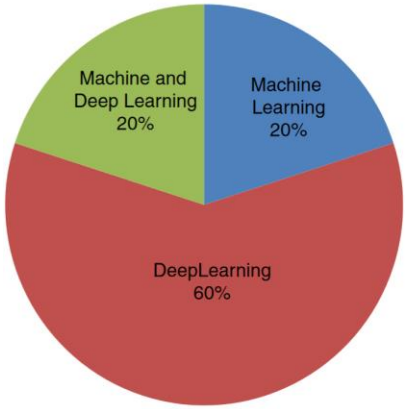
- Accuracies of proposed solutions fall for newer datasets
- No obvious winning methods
- We can use popular models as guideline
- Be cautious when comparing accuracies

Current State of Methods Cont.

Frequency of Machine Learning / Deep Learning Algorithms Used

[4] Z. Ahmad, "Number of time machine learning and deep learning algorithms used"

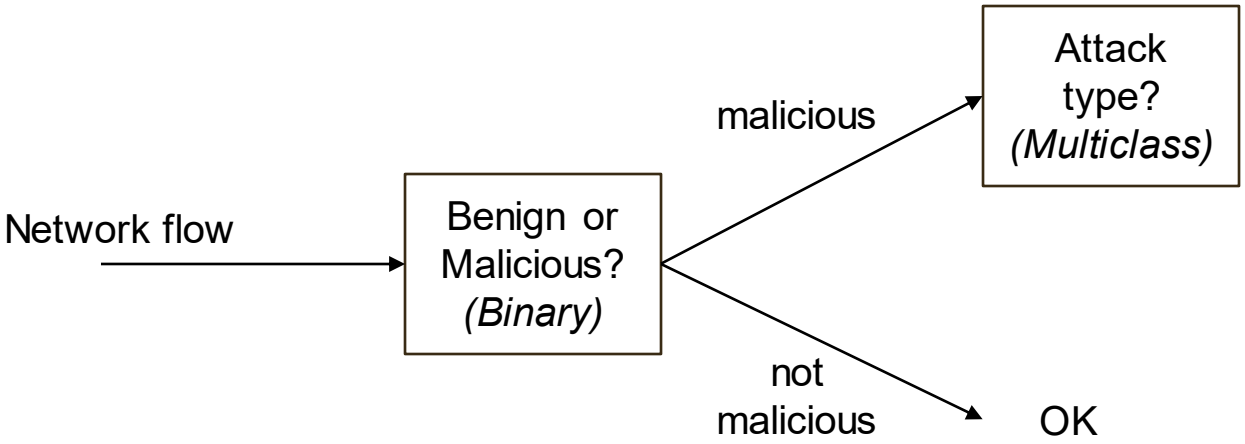




[4] Z. Ahmad, "Methodology distribution"

- Deep learning models are trending for NIDS
  - Better at capturing complex patterns, detecting minor classes
  - Helps automate feature extraction
  - Beware of complexity vs real-world performance

Building our IDS Solution



Our 2 Stage IDS

- A focus on binary classification, as it is more important
- Preliminary results from multiclass classification

Data Preprocessing

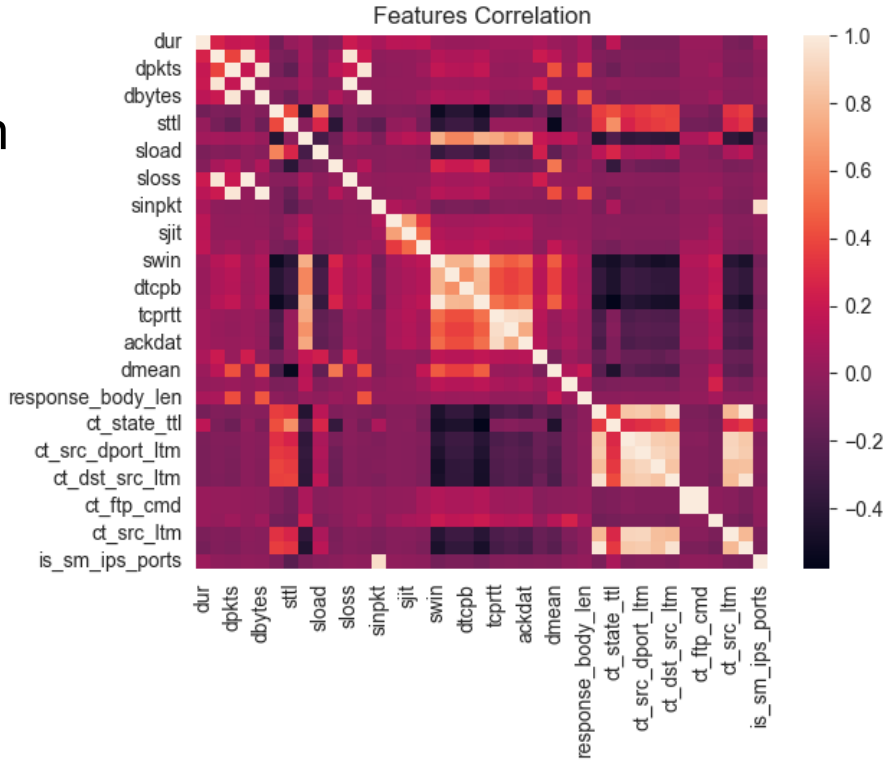
Problem: Training and testing data come from different time periods

- We combined the two data sets then manually split the samples into training and test sets
- Experimental Methods to take into account the time shift:
  - RNN
  - Online learning

Data Preprocessing

Problem: Correlated features

Remove one of every pair of features with higher than 98% correlation



Data Preprocessing

Problem: Categorical Variables

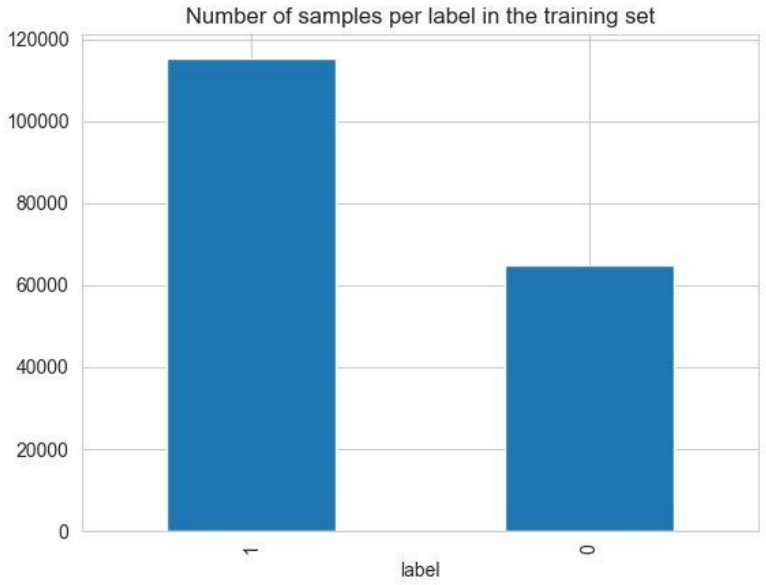
Methods of encoding categorical variables:

- Label Encoding: Assign an integer between 0 and number of categories
- One-Hot Encoding: Add a column for each category
- **Binary Encoding:** Use a binary number to represent each category

Data Preprocessing

Problem: Uneven class distribution

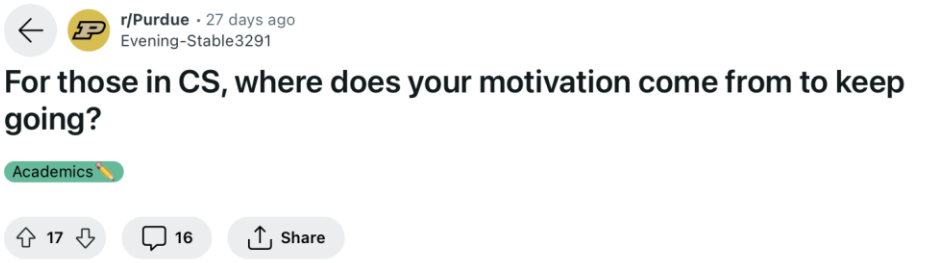
Use class weights to even out the impact of the training samples of each class



Evaluation Criteria

Evaluation Metrics

- Accuracy
- False positives vs false negatives/ confusion matrix
- Precision (False positive higher concern than false negatives)
- Recall (False positive higher concern than false negatives)
- F1 score
- AUC ROC (TPR/Recall vs FPR)



Deep Learning

Layer (type)	Output Shape	Param #
dense_15 (Dense)	(None, 51)	2,652
dropout_8 (Dropout)	(None, 51)	0
dense_16 (Dense)	(None, 51)	2,652
dropout_9 (Dropout)	(None, 51)	0
dense_17 (Dense)	(None, 12)	624
dense_18 (Dense)	(None, 12)	156
dense_19 (Dense)	(None, 1)	13
activation_4 (Activation)	(None, 1)	0

DNN with recombined data:
Acc: 0.9041 F1: 0.9015

With original train/test set:
Acc: 0.8876 F1: 0.9037

Layer (type)	Output Shape	Param #
lstm_24 (LSTM)	(None, 50, 40)	6,720
dropout_33 (Dropout)	(None, 50, 40)	0
lstm_25 (LSTM)	(None, 50, 30)	8,520
dropout_34 (Dropout)	(None, 50, 30)	0
lstm_26 (LSTM)	(None, 50, 30)	7,320
dropout_35 (Dropout)	(None, 50, 30)	0
lstm_27 (LSTM)	(None, 20)	4,080
dropout_36 (Dropout)	(None, 20)	0
dense_21 (Dense)	(None, 64)	1,344
dropout_37 (Dropout)	(None, 64)	0
dense_22 (Dense)	(None, 1)	65
activation_9 (Activation)	(None, 1)	0

RNN with original train/test set:

Acc: 0.8884 F1: 0.9003

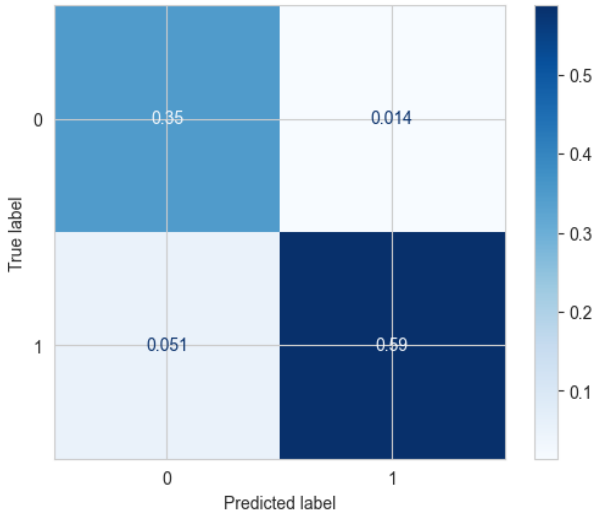
Ensemble Methods

- Most popular methods for this dataset in existing literature

Random Forest:

Fits numerous decision trees on subsamples of the training data

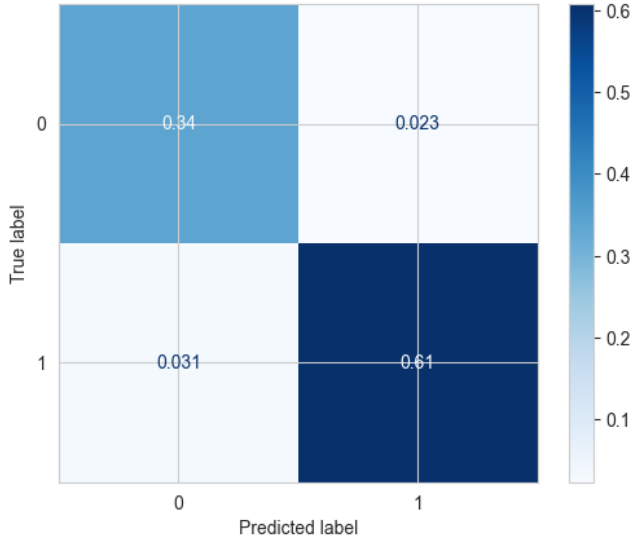
Acc: 0.9350 F1: 0.9477



XGBoost:

Type of boosting ensemble, more resistant to noisy data than Adaboost

Acc: 0.9467 F1: 0.9580

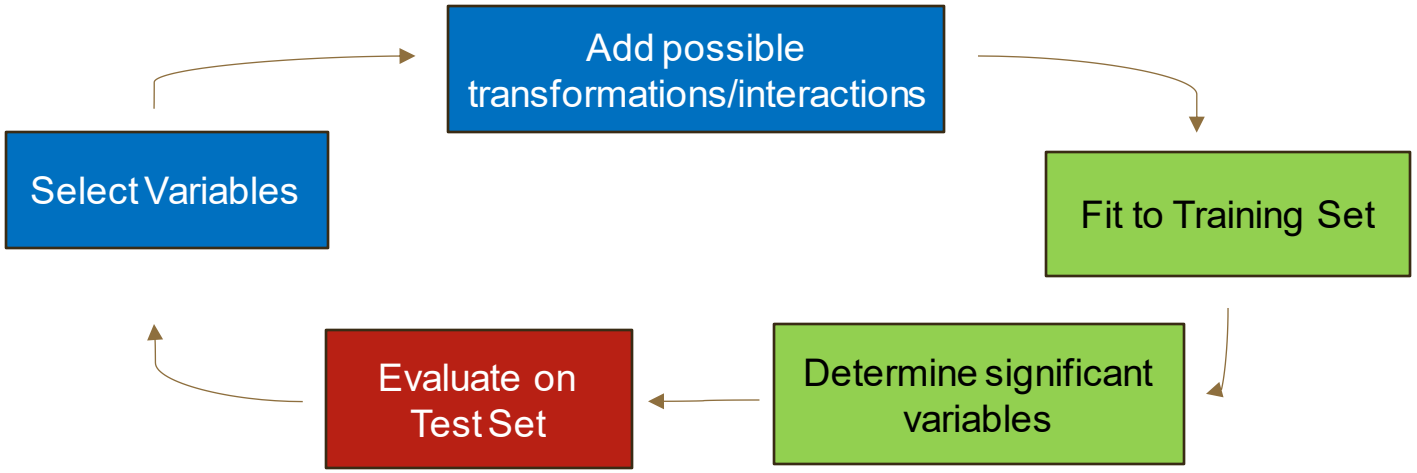


Logistic Regression

Uses of Logistic Regression

- Probabilistic Model
- Parameter Estimation
  - Helps explainability!

Developing Logistic Regression Model



Logistic Regression Cont.

Model 1: label ~ all predictors (-7 correlated predictors)

Balanced Acc: .8450 Sens: .824 Spec: .866

Model 2: label ~ Model 1 predictors + "network bytes"

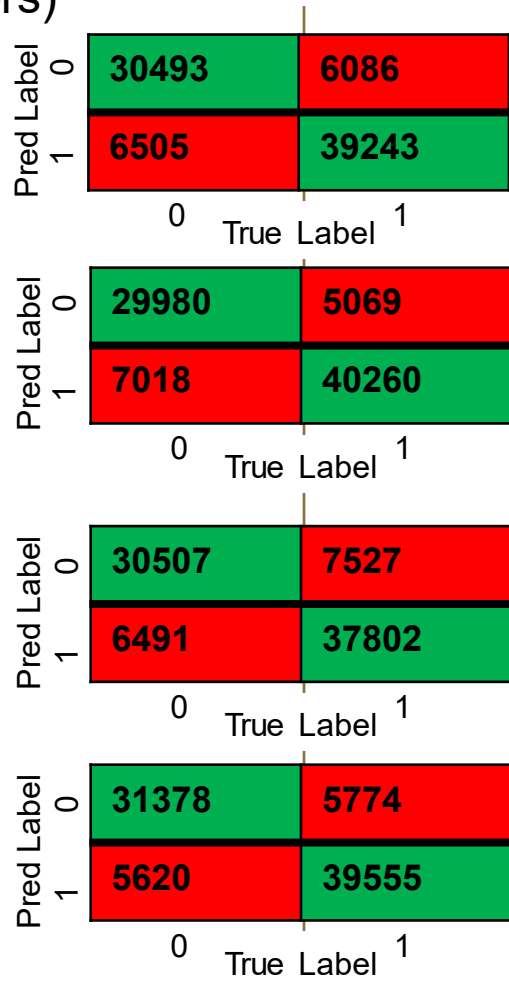
Balanced Acc: .8492 Sens: .810 Spec: .889

Model 3: label ~ categorical predictors + 6 most significant numerical predictors

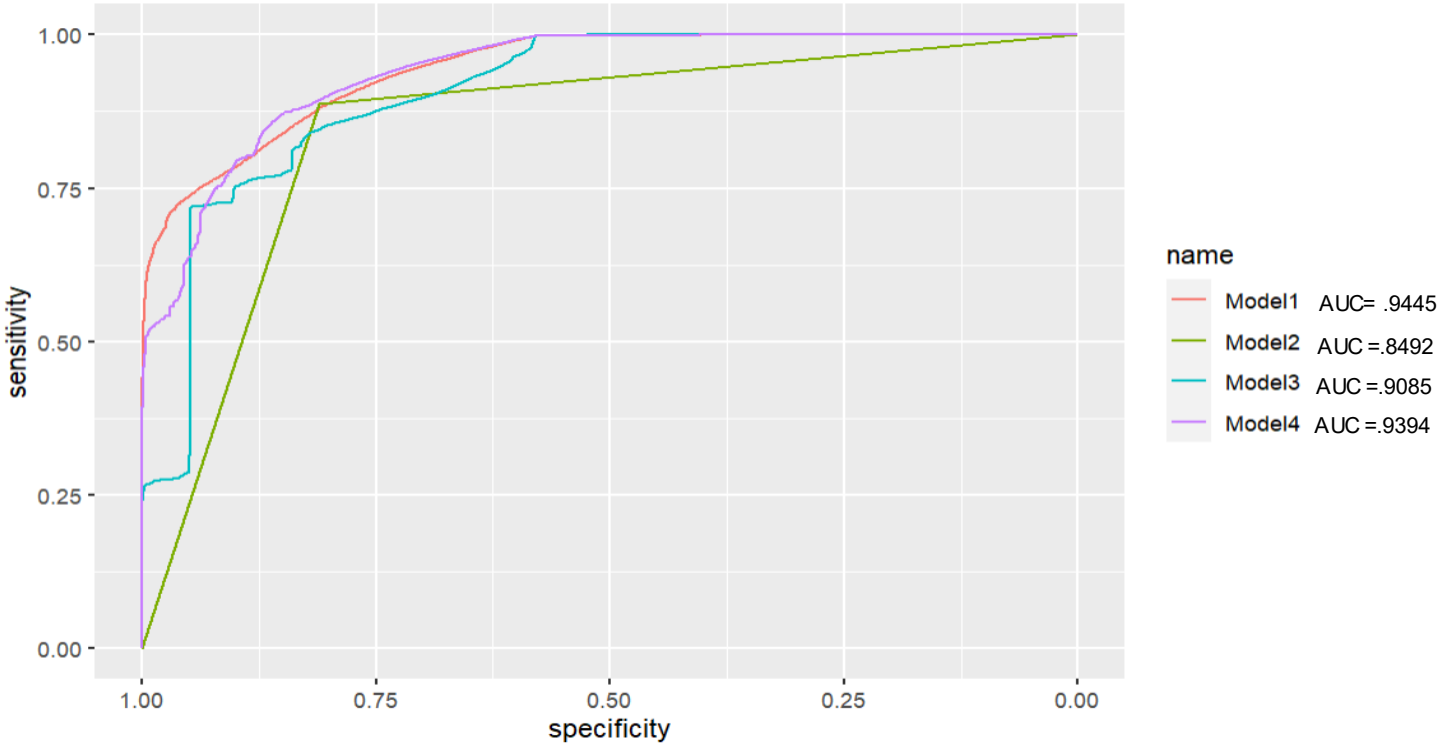
Balanced Acc: .8293 Sens: .825 Spec: .834

Model 4: label ~ categorical predictors + 15 most significant numerical predictors

Balanced Acc: .8604 Sens: .8481 Spec: .873



LR Model Comparison Cont.



Combination of Unsupervised and Supervised Learning

Methodology and Evaluation

- Train unsupervised clustering method and use to predict if malicious or not.
- Create new feature in training and testing sets with predictions from unsupervised model.
- Evaluate to see if this new information increases accuracy when using the supervised learning method.
- This is a form of semi-supervised learning

Unsupervised Method	Accuracy	F1 Score
Original Model	0.9041	0.9015
Gaussian Mixture	0.9157	0.9308
Birch	0.9124	0.9278
K-Means	0.9065	0.9271
Ensemble	0.9273	0.9424

Results – Binary Classification

	Model	Test Accuracy	F1 Score
Supervised	Logistic Regression	0.8604	0.8616
	Random Forest	0.9350	0.9477
	XGBoost	0.9467	0.9580
	NN	0.9041	0.9219
	RNN (uncombined data)	0.8884	0.9003
Semi-supervised	Gaussian Mixture	0.9157	0.9308
	Birch	0.9124	0.9278
	K-Means	0.9065	0.9271
	Ensemble	0.9273	0.9424

Preliminary Results – Multiclass Classification

	Model	Test Accuracy	Weighted F1 Score
Supervised	NN	0.6493	0.7012
	Random Forest	0.7972	0.8164
	XGBoost	0.8351	0.8179

Before the end of the semester:

- Apply all previously used techniques on multiclass classification
- Experiment with other data preprocessing techniques such as PCA
- Apply online learning techniques to handle the data shift over time

Exploration of Online Learning

How does it work?

- Minibatch and continuous learning
- Incremental updates
- Tackles data as it arrives
- Allows for real-time analysis and predictions

Do real-time predictions work?

- Yes, but with some caveats.
- Very promising results so far.
- How to combine batch learning and online learning?

Classification Task	Model Used	Accuracy
Binary	Adaptive Logistic Regression	.8763
	Adaptive Random Forest	.9337
Multiclass	Hoeffding Trees	.7196
	OneVsOne Classifier	.7244

Conclusion

- Online learning appears very promising
- Previous works failed to address the time shift
- Ensemble methods generally perform the best
- False Alarm rate is still a concern
- Optimism for new methods and techniques to increase performance on difficult data sets such as these

Works Cited

- [1] - https://www.forbes.com/advisor/education/it-and-tech/cybersecurity-statistics/
- [2] - https://www.infosecinstitute.com/resources/general-security/cyber-espionage-the-greatest-transfer-of-wealth-in-history/#:~:text=Cyber%20espionage%20Statistics&text=NSA%20Director%20General%20Keith%20Alexander,a%20further%20%24114%20billion%20annually.
- [3]- https://community.trustcloud.ai/article/security-meme-100-funny-cyber-security-memes-compliance-memes-2024/
- [4] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," Trans Emerging Tel Tech, vol. 32, no. 1, p. e4150, Jan. 2021, doi: 10.1002/ett.4150.
- [5] P. Vanin *et al.*, "A Study of Network Intrusion Detection Systems Using Artificial Intelligence/Machine Learning," *Applied Sciences*, vol. 12, no. 22, p. 11752, Nov. 2022, doi: 10.3390/app122211752.
- [6] S. More, M. Idrissi, H. Mahmoud, and A. T. Asyhari, "Enhanced Intrusion Detection Systems Performance with UNSW-NB15 Data Analysis," *Algorithms*, vol. 17, no. 2, p. 64, Feb. 2024, doi: 10.3390/a17020064.

THANK YOU

Questions?

Heart Disease Classification

CS573 Project

Haoze Li/ Xueyuan Cao/ Yuan Zhou/ Qinglin Meng

Background & Exploratory Data Analysis

Heart Attack Possibility (Haoze Li start)

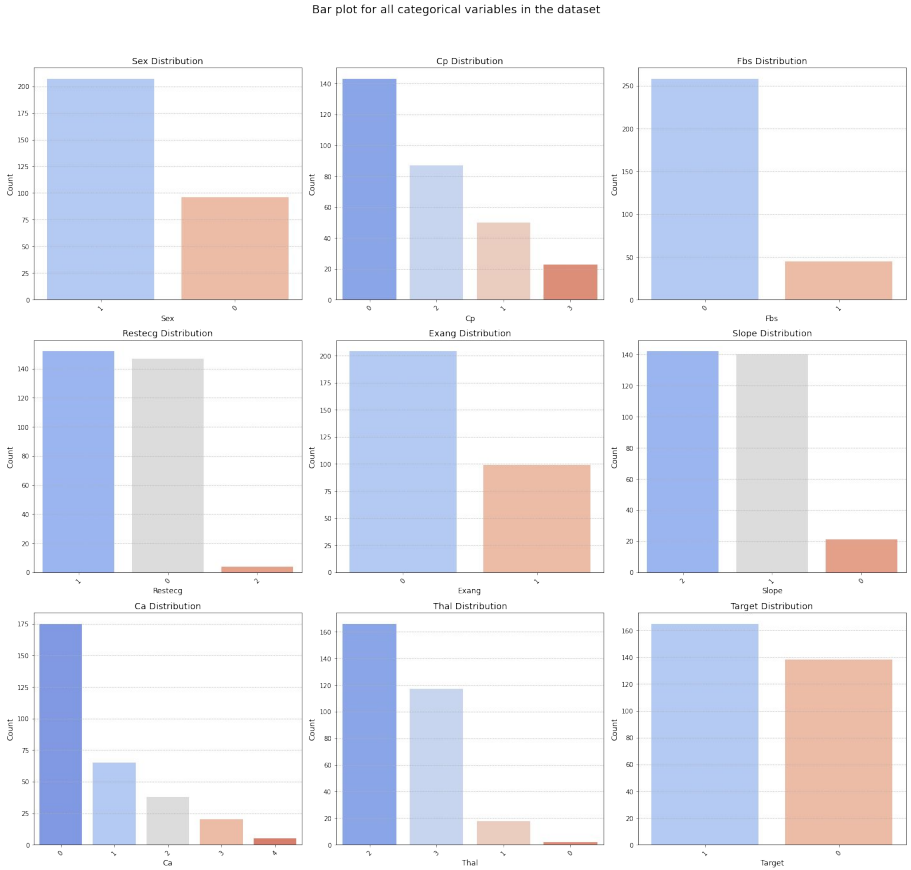
- This dataset contains 14 attributes. The target variable refers to the presence of heart disease in the patient.
- Some descriptive statistics can be found below:

	count	mean	std	min	25%	50%	75%	max
age	303.0	54.366337	9.082101	29.0	47.5	55.0	61.0	77.0
sex	303.0	0.683168	0.466011	0.0	0.0	1.0	1.0	1.0
cp	303.0	0.966997	1.032052	0.0	0.0	1.0	2.0	3.0
trestbps	303.0	131.623762	17.538143	94.0	120.0	130.0	140.0	200.0
chol	303.0	246.264026	51.830751	126.0	211.0	240.0	274.5	564.0
fbs	303.0	0.148515	0.356198	0.0	0.0	0.0	0.0	1.0
restecg	303.0	0.528053	0.525860	0.0	0.0	1.0	1.0	2.0
thalach	303.0	149.646865	22.905161	71.0	133.5	153.0	166.0	202.0
exang	303.0	0.326733	0.469794	0.0	0.0	0.0	1.0	1.0
oldpeak	303.0	1.039604	1.161075	0.0	0.0	0.8	1.6	6.2
slope	303.0	1.399340	0.616226	0.0	1.0	1.0	2.0	2.0
ca	303.0	0.729373	1.022606	0.0	0.0	0.0	1.0	4.0
thal	303.0	2.313531	0.612277	0.0	2.0	2.0	3.0	3.0
target	303.0	0.544554	0.498835	0.0	0.0	1.0	1.0	1.0

Categorical variables (9):
1. sex
2. cp *(chest pain type)*
3. fbs *(fasting blood sugar)*
4. restecg *(resting electrocardiographic results)*
5. exang *(exercise induced angina)*
6. slope *(the slope of the peak exercise ST segment)*
7. ca *(number of major vessels colored by fluoroscopy)*
8. thal *(type of defect)*
9. target (if the patient has more chance of heart attack

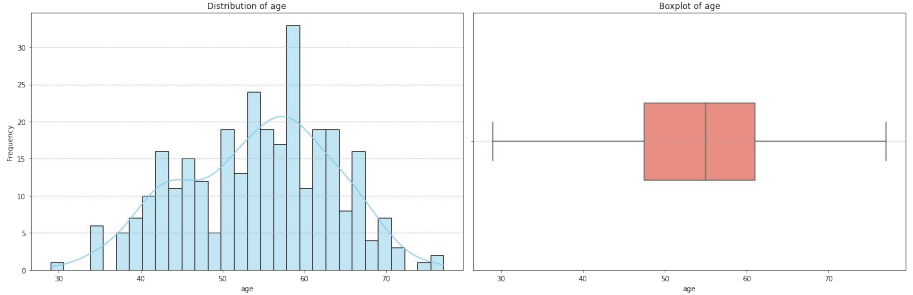
Numerical variables (5):
1. age
2. trestbps *(resting blood pressure)*
3. chol *(serum cholestoral in mg/dl)*
4. thalach *(maximum heart rate achieved)*
5. oldpeak *(ST depression induced by exercise relative to rest)*

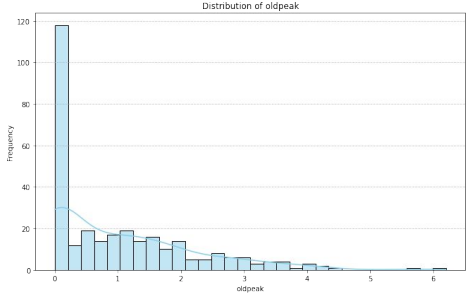
Bar Plot for Categorical Variables

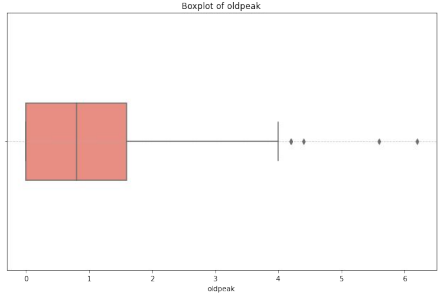


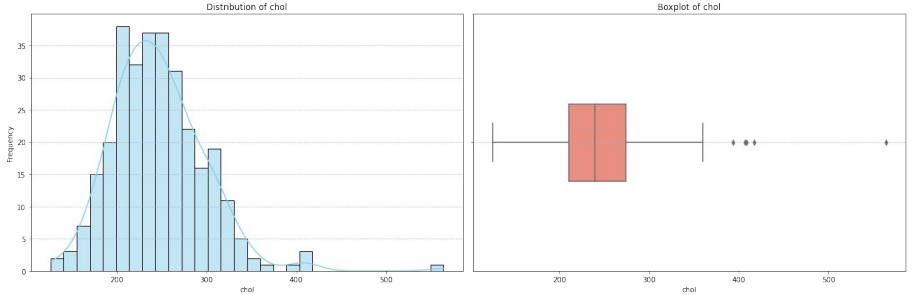
The target is balanced!

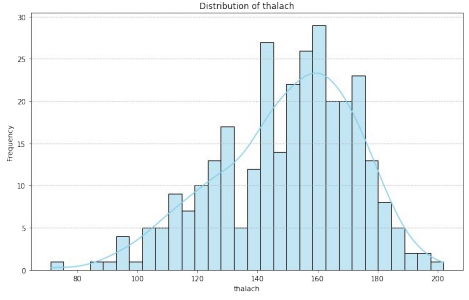
Distribution of Numerical Variables

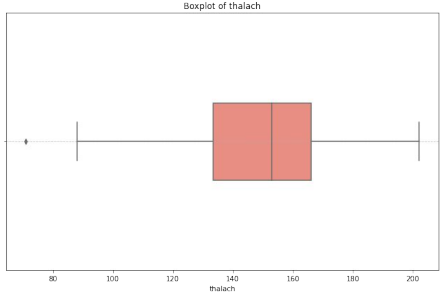


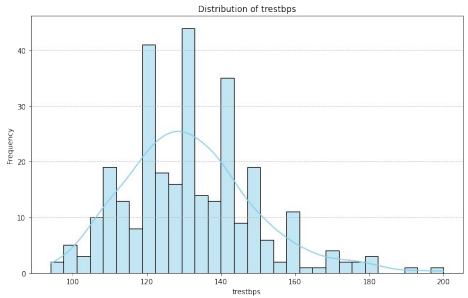


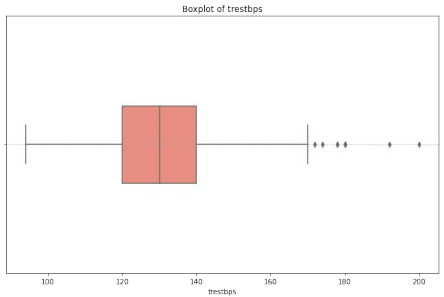




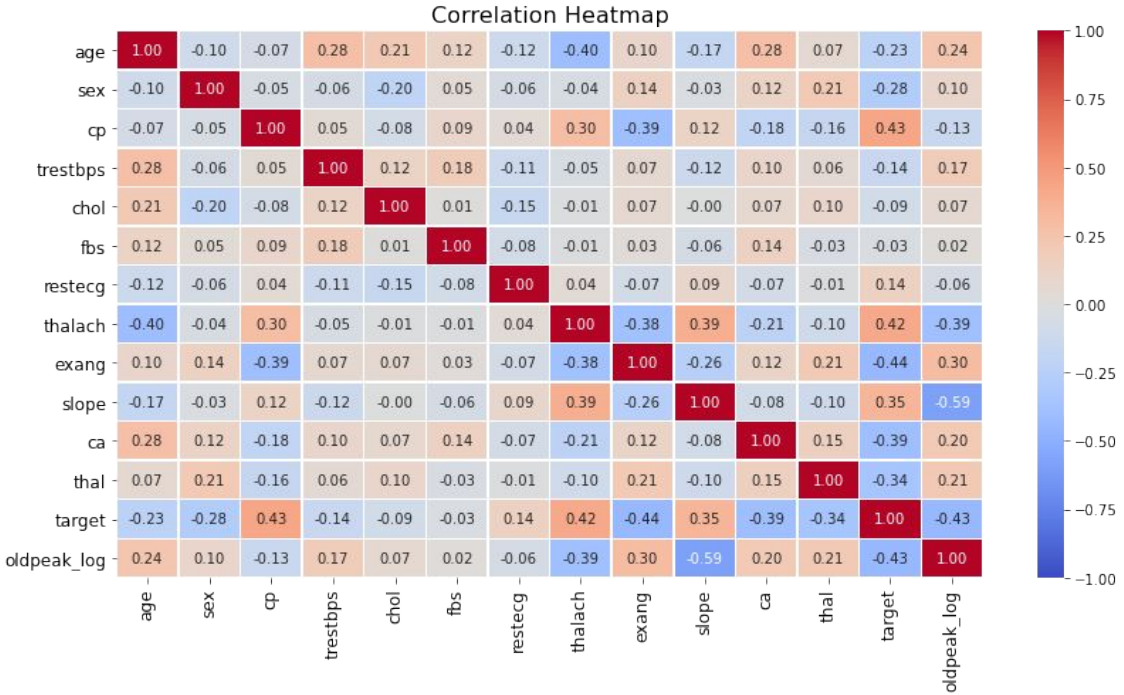








Correlation Hotmap



No strongly correlated variables

Methods

kNN - Introduction (Xueyuan Cao start)

- Fundamental, non-parametric method
- Basic principle:
  - Find the k closest data points in the feature space to a given input point and predicts the output based on these neighboring points
- Boosting performance:
  - k-fold cross-validation (k = [3, 5, 10])
  - PCA (n_components = 0.95)

kNN - Applying

In order to find out whether k-fold cross-validation is boosting the performance, we compared the results of knn-k-fold, knn–k-fold-pca, knn, knn-pca

```
Running k-fold with 3 folds
Running k-fold with 5 folds
Running k-fold with 10 folds
Best k: 5, Best Neighbor: 8, Best Accuracy: 0.676031746031746
Time Taken: 0.21927704199333675
```

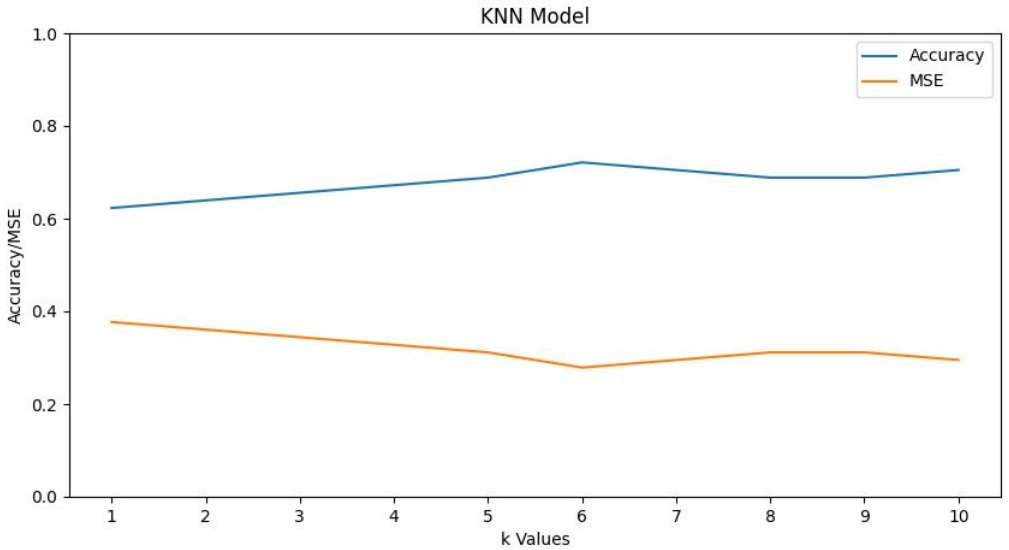
```
Running k-fold with 3 folds
Running k-fold with 5 folds
Running k-fold with 10 folds
Best k: 3, Best Neighbor: 10, Best Accuracy: 0.8448844884488449
Time Taken: 0.23712766700191423
```

```
Without Standardization and Dimensionality Reduction:
>> Neighbors: 1, Accuracy: 0.6229508196721312, MSE: 0.3770491803278688
>> Neighbors: 2, Accuracy: 0.639344262295082, MSE: 0.36065573770491804
>> Neighbors: 3, Accuracy: 0.6557377049180327, MSE: 0.3442622950819672
>> Neighbors: 4, Accuracy: 0.6721311475409836, MSE: 0.32786885245901637
>> Neighbors: 5, Accuracy: 0.6885245901639344, MSE: 0.3114754098360656
>> Neighbors: 6, Accuracy: 0.7213114754098361, MSE: 0.2786885245901639
>> Neighbors: 7, Accuracy: 0.7049180327868853, MSE: 0.29508196721311475
>> Neighbors: 8, Accuracy: 0.6885245901639344, MSE: 0.3114754098360656
>> Neighbors: 9, Accuracy: 0.6885245901639344, MSE: 0.3114754098360656
>> Neighbors: 10, Accuracy: 0.7049180327868853, MSE: 0.29508196721311475
Time Taken: 0.03644974998314865
```

```
With Standardization and Dimensionality Reduction:
>> Neighbors: 1, Accuracy: 0.8524590163934426, MSE: 0.14754098360655737
>> Neighbors: 2, Accuracy: 0.8032786885245902, MSE: 0.1967213114754098
>> Neighbors: 3, Accuracy: 0.8524590163934426, MSE: 0.14754098360655737
>> Neighbors: 4, Accuracy: 0.8360655737704918, MSE: 0.16393442622950818
>> Neighbors: 5, Accuracy: 0.8852459016393442, MSE: 0.11475409836065574
>> Neighbors: 6, Accuracy: 0.8688524590163934, MSE: 0.13114754098360656
>> Neighbors: 7, Accuracy: 0.8688524590163934, MSE: 0.13114754098360656
>> Neighbors: 8, Accuracy: 0.8688524590163934, MSE: 0.13114754098360656
>> Neighbors: 9, Accuracy: 0.9016393442622951, MSE: 0.09836065573770492
>> Neighbors: 10, Accuracy: 0.9016393442622951, MSE: 0.09836065573770492
Time Taken: 0.030695167020894587
```

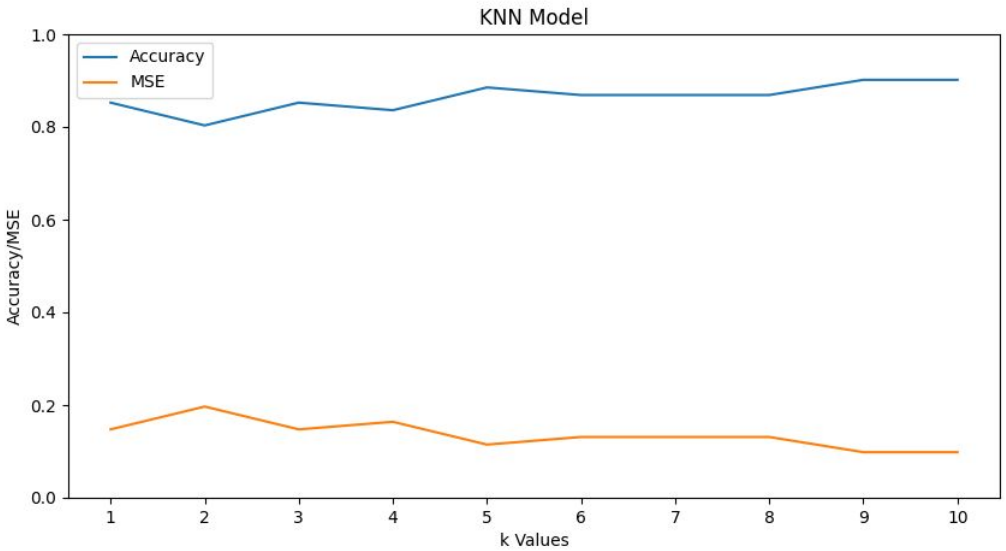
kNN-Applying (cont.)

- Through kNN to the data, we could get the following detailed data.
- As we could see from the graph, the accuracy of the kNN model is roughly 0.6~0.7.



kNN-Applying (cont.)

- However, due to this architecture of this model, it can, its performance can be significantly affected by the curse of dimensionality and the need for meticulous data preprocessing.
- Based on the above understanding, we can propose a simple yet affective improvement to our base kNN model - PCA
- After applying PCA, we could see that the accuracy was boosted to 0.9.

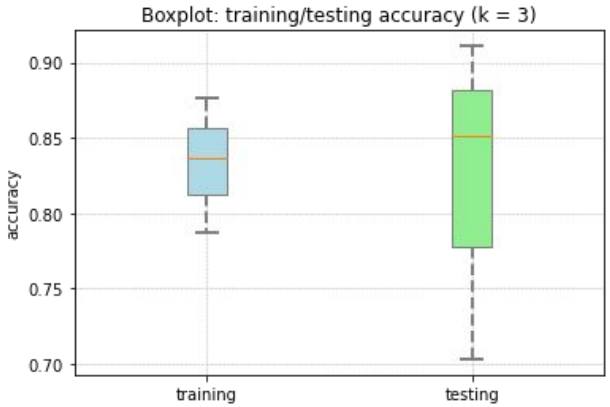


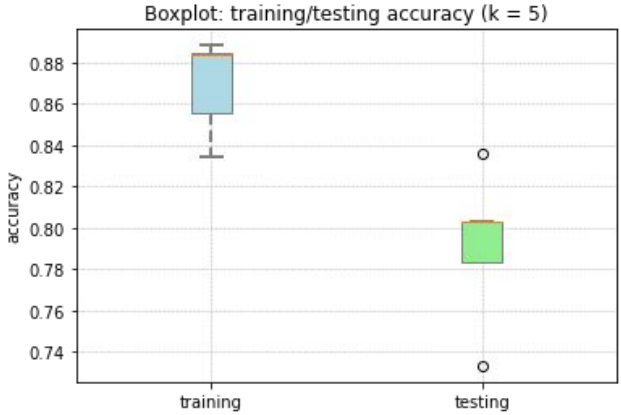
Neural Network (NN) - Introduction

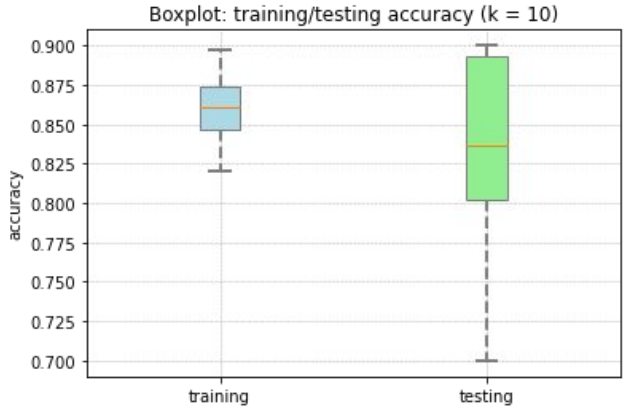
- Inspired by the human brain’s structure and function
- The heart of NN: nodes (or neurons)
  - processes incoming data and passes its output to subsequent layers
- NN learns by adjusting the weights of connections between neurons based on the errors in predictions, a process facilitated by algorithms such as backpropagation
- Our Neural network data:
  - Layers: 5
  - Activations: Tanh (hyperbolic tangent function)
  - Optimizer: Adam
  - Epochs: 500
  - Learning rate: 1e-3

Neural Network - applying

- In order to improve the performance of NN, we applied k-fold cross validation.
- Based on a simple observation, we could see that the after applying k = 3, 5, 10 fold cross-validation, the accuracy was boosted to around 0.9







Logistic Regression - Introduction (Yuan Zhou start)

- Logistic function

$$f(z) = \frac{1}{1 + e^{-z}}$$

- Kernel logistic regression
  - Radial Basis Function (RBF) kernel

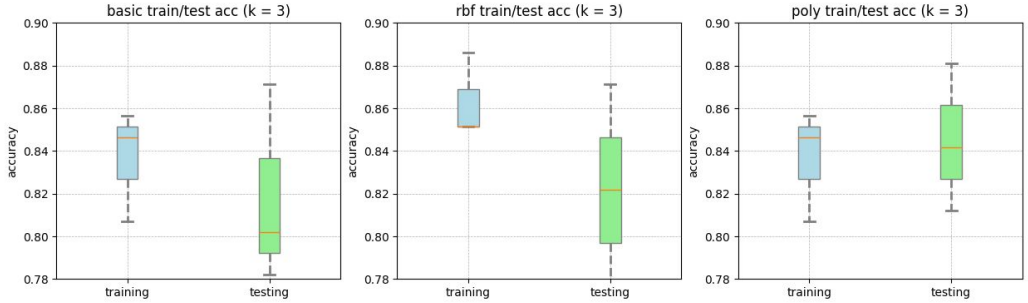
$$K(x, x') = \exp(-\gamma ||x - x'||^2)$$

  - Polynomial Kernel (d=2)

$$K(x, x') = (1 + x \cdot x')^d$$

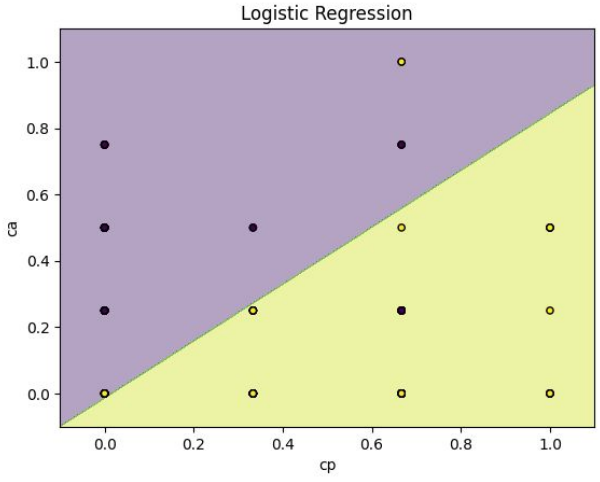
- 3-fold cross validation

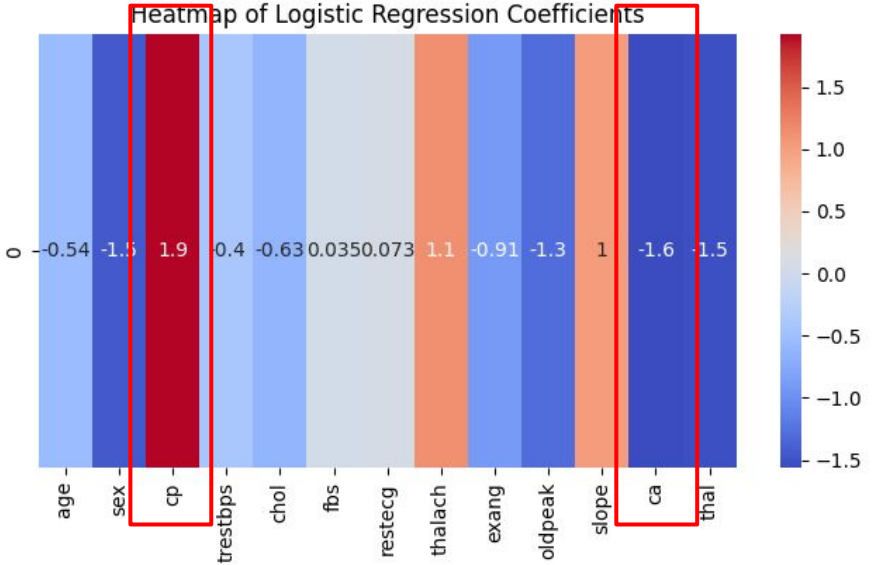
	Basic	RBF	Polynomial
train_acc	0.84	0.86	0.88
test_acc	0.82	0.82	**0.85**



Logistic Regression - Applying

- Basic LR coefficients heatmap
- Decision boundary





Gaussian Naive Bayes - Introduction

- Bayes' Theorem

$$P(y|x_1, ..., x_n) = \frac{P(x_1, ..., x_n|y)P(y)}{P(x_1, ..., x_n)}$$

- Assumption of feature conditionally Independent given class y

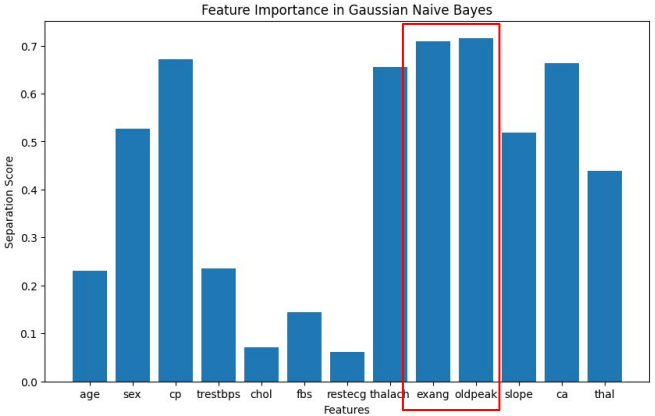
$$P(x_1, ..., x_n|y) = \prod_{i=1}^{n} P(x_i|y)$$

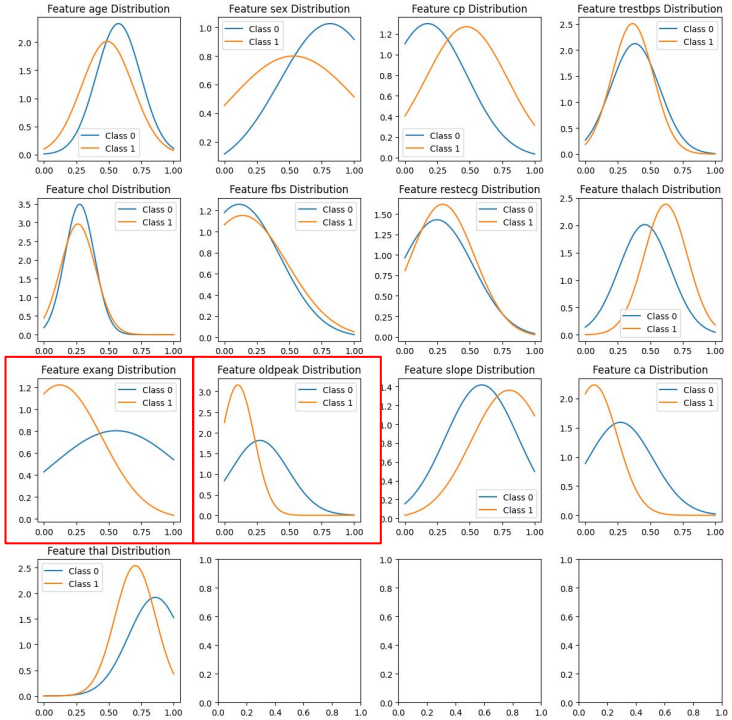
- Gaussian Distribution of Features: feature x_i is assumed to follow a Gaussian distribution conditional on the class y

Gaussian Naive Bayes - Applying

- 3-fold cross validation
  - Train acc: 0.84; Test acc: 0.82
- Calculate the separation scores for each feature

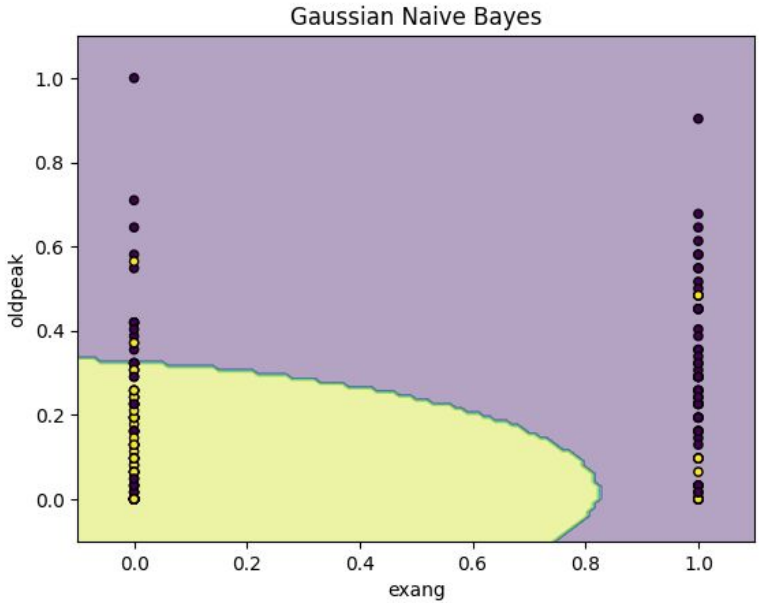
$$S_j = \frac{|\mu_{j,c_1} - \mu_{j,c2}|}{\sqrt{\sigma^2_{j,c_1} + \sigma^2_{j,c_2}}}$$





Gaussian Naive Bayes - Applying

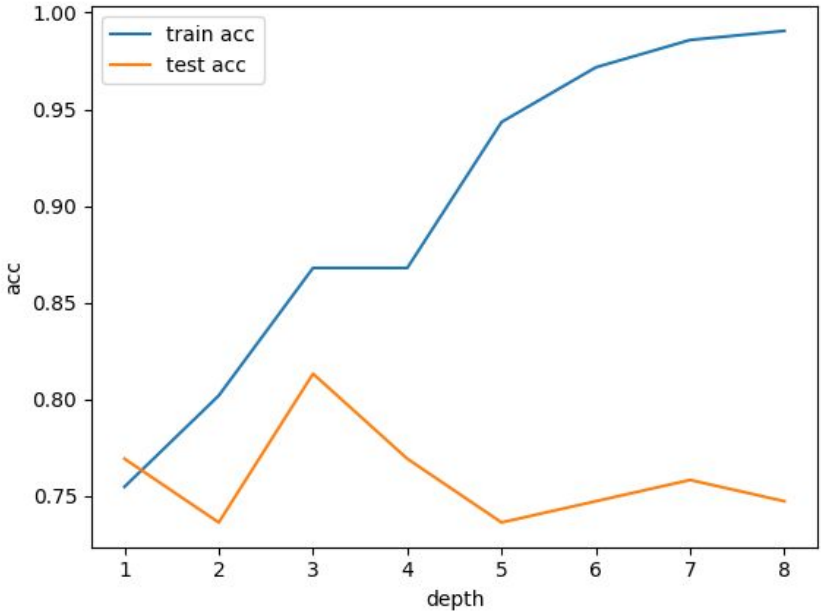
- Plot decision boundary



Decision Tree - Introduction (Qinglin Meng start)

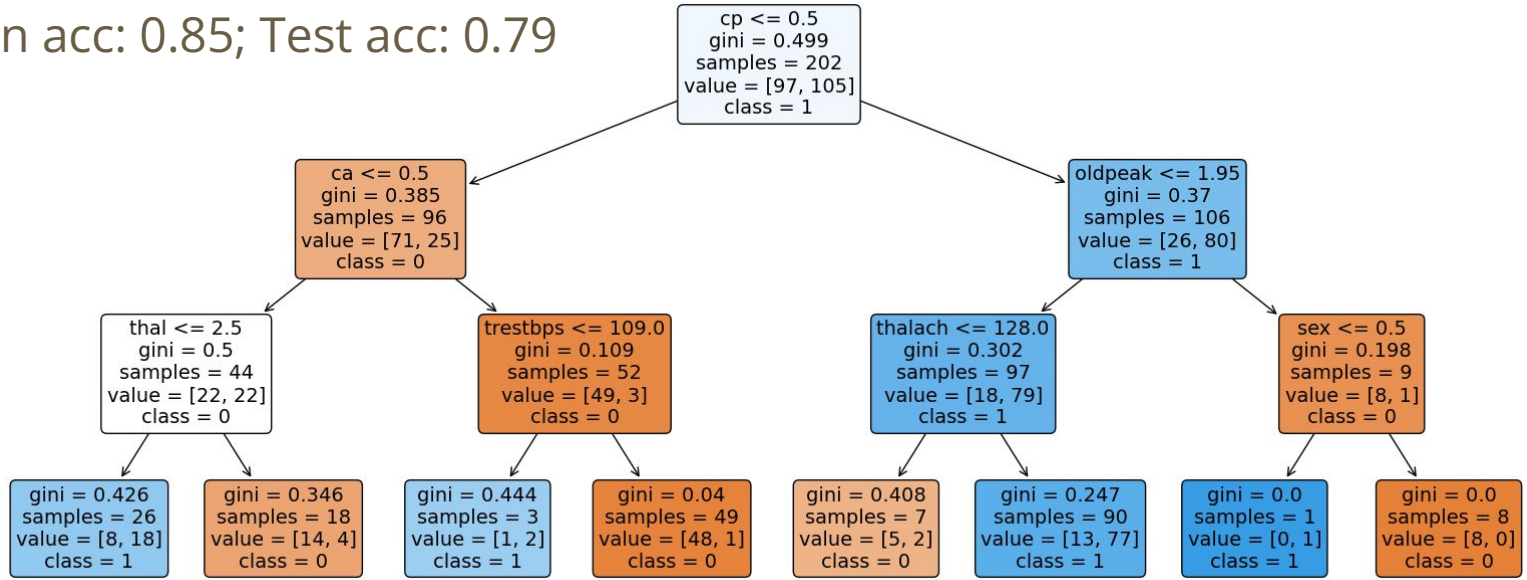
- A tree-like model
- Nodes: represents a decision point or a condition
- Branches: represent the possible outcomes leading to further nodes or to final decisions
- Chose depth (hyper-parameter)

select 3!



Decision Tree - Applying

- Set depth=3
- Train acc: 0.85; Test acc: 0.79
-



Decision Tree-Adaboosting

- n=2

  Train acc = 0.89, Test acc = 0.78

- n=5

  Train acc = 0.96, Test acc = 0.78

  Overfitting!

Conclusion

Model Comparison

1. Test Accuracy

KNN	Neural Network	Poly-LR	Naive Bayes	Decision Tree
0.84	0.80	**0.85**	0.82	0.79

2. Test Sensitivity and Specificity (Sensitivity is more important!)

	KNN	Neural Network	**Poly–LR**	Decision Tree
Sensitivity	0.87	0.86	**0.89**	0.78
Specificity	0.80	0.76	0.79	0.78

Factors Causing Heart Attact

- Chest Pain (CP)
- Number of major vessels (0-3) colored by flourosopy (CA)
- ST depression induced by exercise relative to rest (Oldpeak)
- …

Induced from Logestic Regression and Decision Tree.

Our results can be used for preliminary screening of heart diseases.

Future Work

1. Dataset is a little bit small (300)

   We can acquire more labeled data.

   We can use active learning methods to get labeled data instead of passive learning methods.

2. Features can be more detailed.

   e.g. chest pain can be specified as more than 4 classes.

Uncovering Conversation Patterns through Data Mining

Zhaoqing Wu

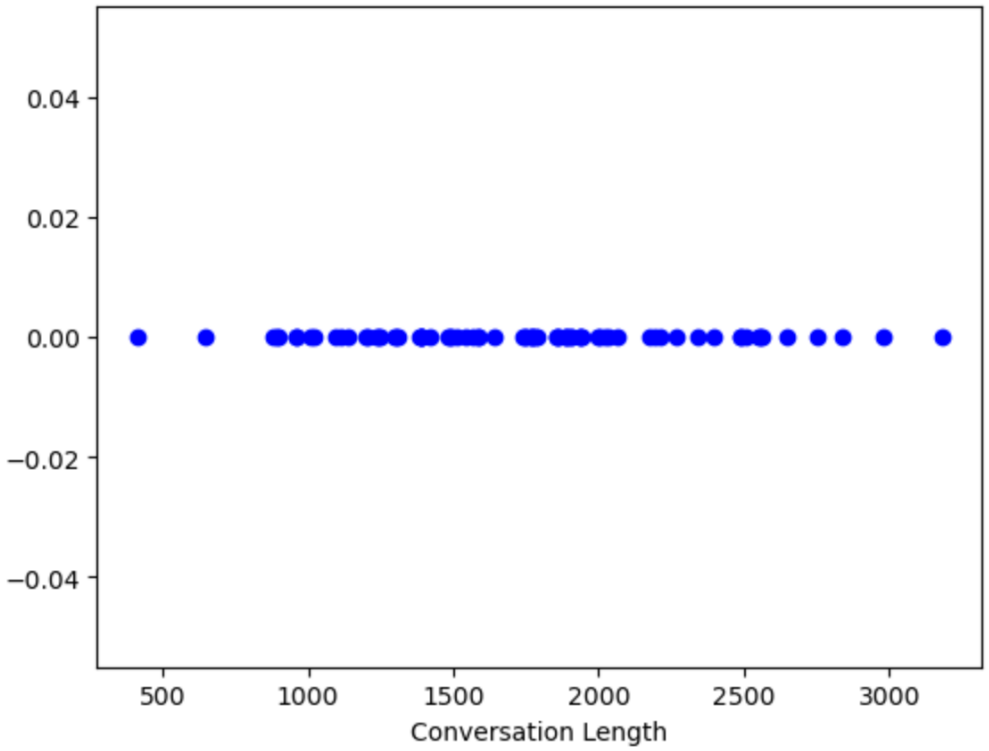
Introduction

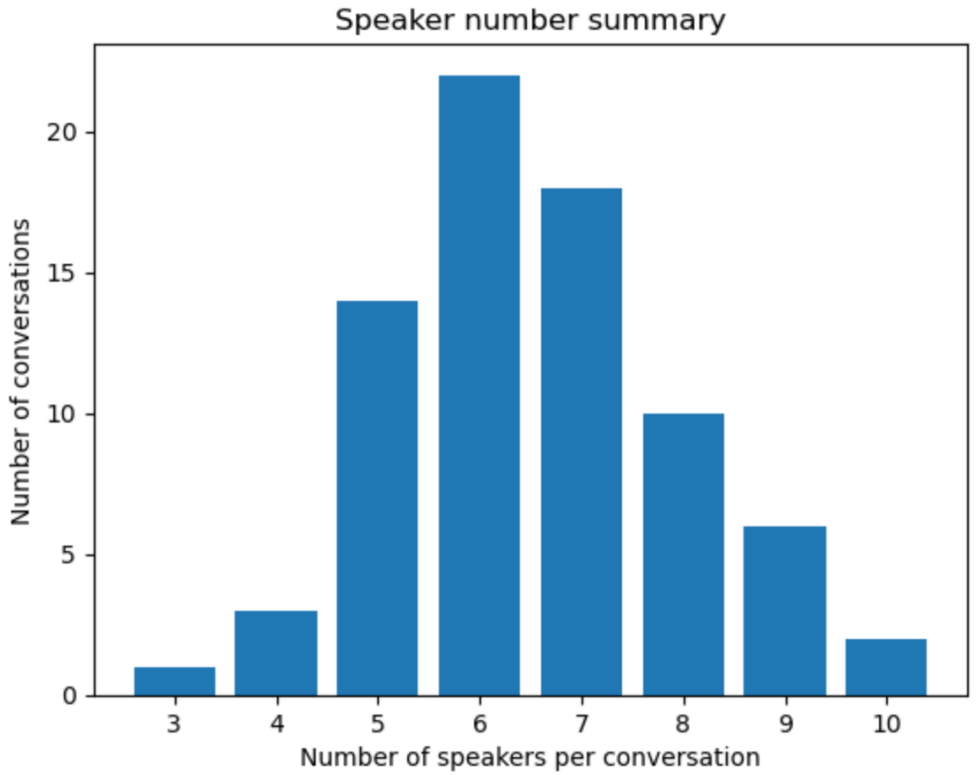
- Conversation patterns
  - Phrases or vocabulary that repeately occur
  - Speech act (Green, 2021)
  - Language coordination (Danescu-Niculescu-Mizil, 2012)
  - Etc.
- Questions
  - How do people speak differently in conversations?
  - How is such difference related to their sociodemographic background?
- Objectives
  - Identify the status of the speaker given the utterance.

Data

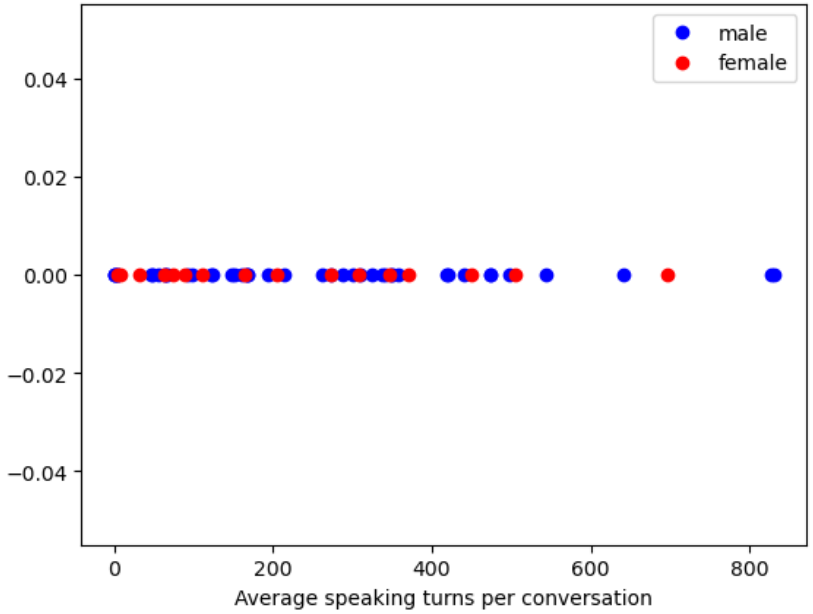
- Natural meetings at the International Computer Science Institute (ICSI), Berkeley (Janin, et al., 2003)
  - 75 meetings
  - Rich transcription
  - Metadata for speakers
    - Roles (professor, grad student, undergrad, etc.)
    - Gender
    - Age
    - Native languages

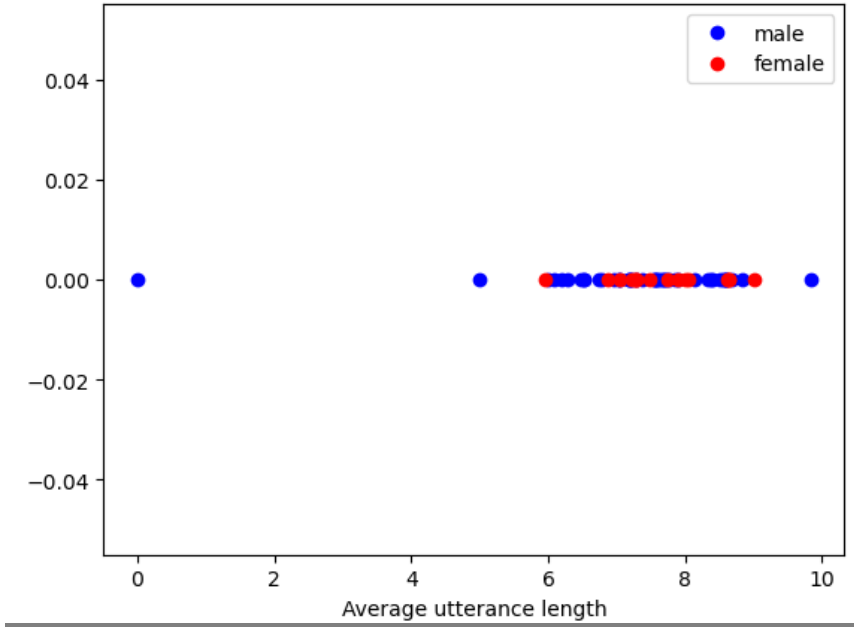
Exploratory Data Analysis

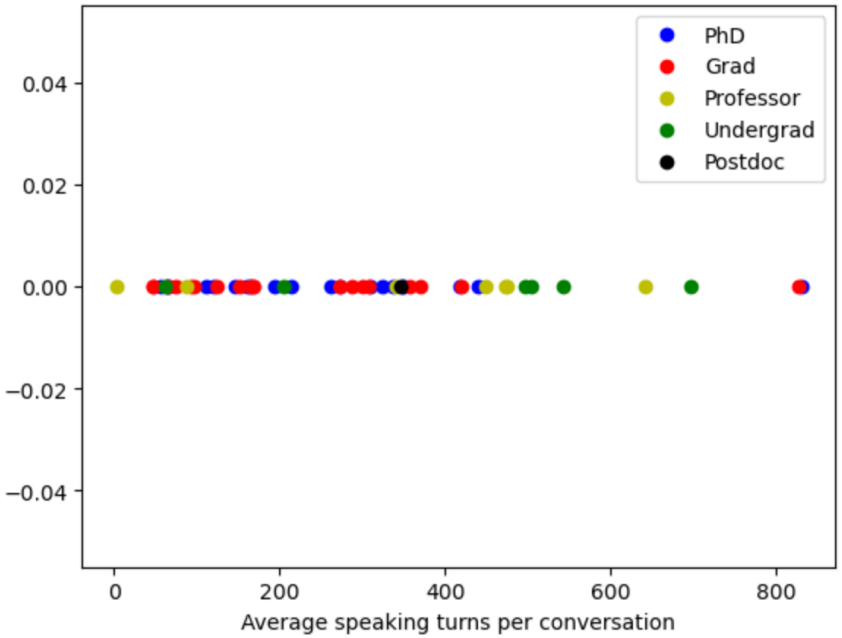


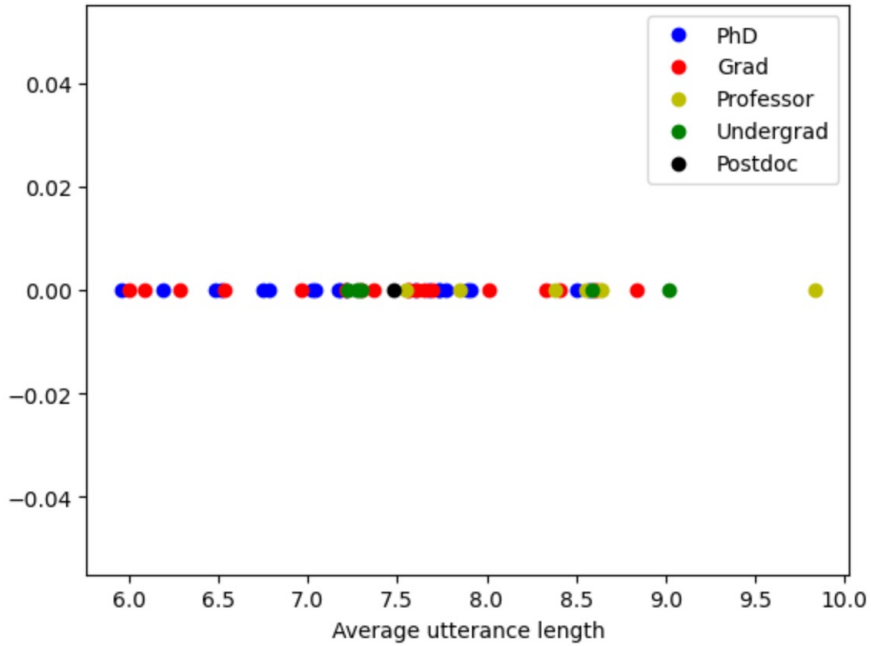


CS 573 Final Project

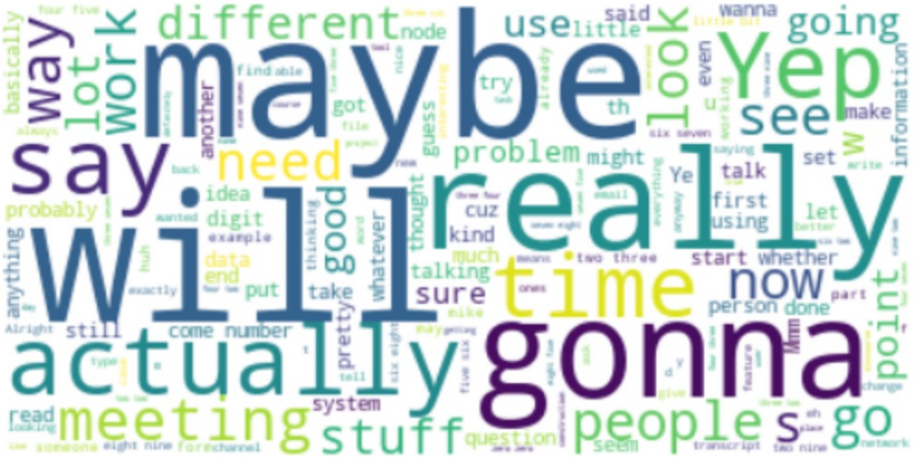








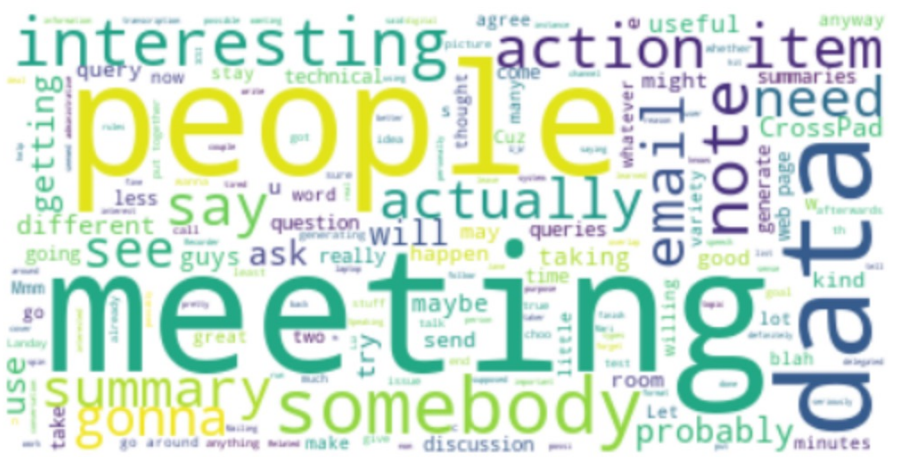
Male Grad



Male Professor

Female Grad

Female Professor



Features

	Male	Female	Professor	Grad Student
Questions	**0.077**	**0.063**	0.081	0.086
Repetitions	0.397	0.420	**0.452**	**0.353**
Interruptions	0.327	0.303	0.366	0.294
Backchannels	0.048	0.052	**0.027**	**0.047**
Short responses	0.044	0.043	**0.025**	**0.041**

Model Evaluation

Gender prediction

	precision	recall	F1 score
Naïve Bayes	76.8	67.5	71.8
SVM	70.8	69.1	69.9

Role prediction

	precision	recall	F1 score
Naïve Bayes	67.2	76.2	71.4
SVM	71.6	73.6	72.6

Further Work

- Incorporating features into predictions.
- Write the report.

References

- Cristian Danescu-Niculescu-Mizil, Lillian Lee, Bo Pang, and Jon Kleinberg. 2012. Echoes of power: language effects and power differences in social interaction. In Proceedings of the 21st international conference on World Wide Web (WWW '12). Association for Computing Machinery, New York, NY, USA, 699–708. https://doi.org/10.1145/2187836.2187931
- Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke and Chunk Woosters. 2003. The ICSI Meeting Corpus. In Proceedings of 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003 (ICASSP ’03). Hong Kong, China, 2003, pp. I-I, doi: 10.1109/ICASSP.2003.1198793.
- Green, Mitchell, "Speech Acts", *The Stanford Encyclopedia of Philosophy* (Fall 2021 Edition), Edward N. Zalta (ed.), URL = <https://plato.stanford.edu/archives/fall2021/entries/speech-acts/>.

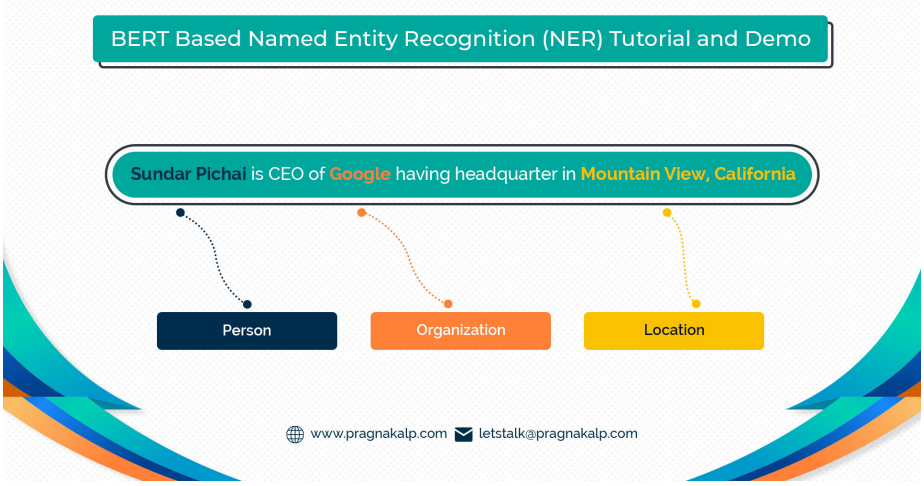
Weakly Supervised Named Entity Tagging with Contrastive Learning

Zhichuan Duan, Yu Lin, Hieu Tran

April 2024

1 What's Named Entity Recognition?

Named Entity Recognition (NER) is a crucial task in natural language processing (NLP) that involves identifying and categorizing named entities within text data. These entities can include people, organizations, geographical locations and dates or times.

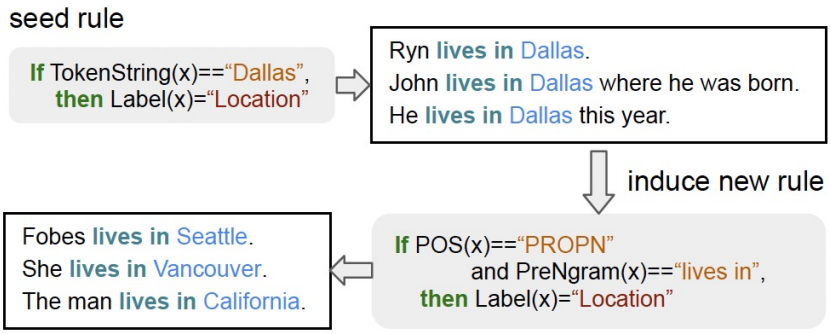


2 TALLOR method

TALLOR[1] can automatically learn new rules from unlabeled data and a small set of seed rules (e.g. 20 rules).

1. J. Li, H. Ding, J. Shang, J. McAuley, Z. Feng, Weakly supervised named entity tagging with learnable logical rules, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the

11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 4568–4581.



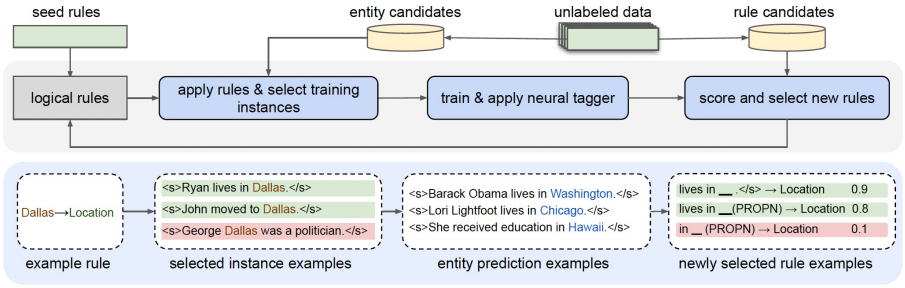
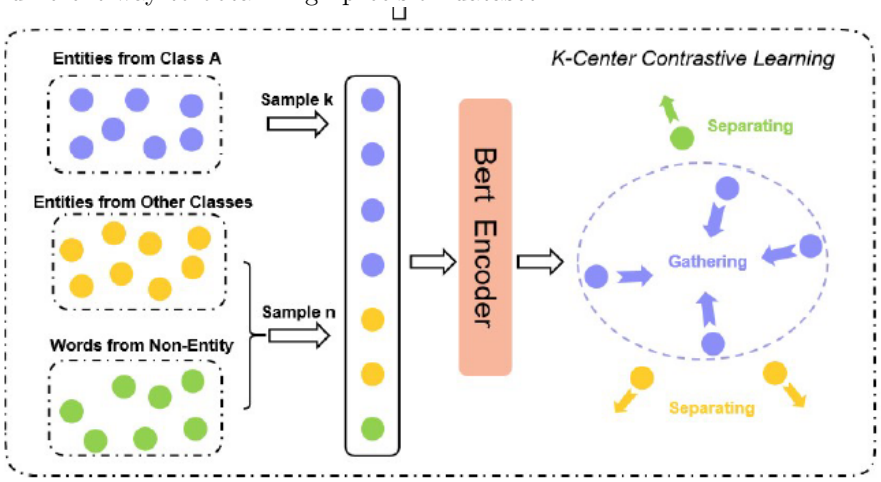


Figure 2: Overview of our tagging framework with logical rules and examples for each step.

3 K-center contrastive learning

A different way to obtain high precision dataset.



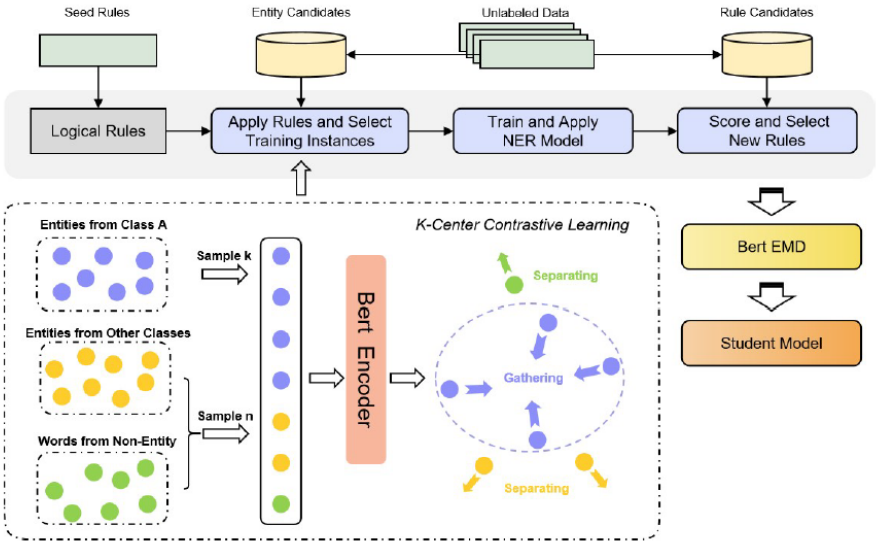
We use high precision dataset, which in our case is the manually labeled data to train the decision boundaries.
The suitable decision boundaries should satisfy two conditions. On the one hand, they should be broad enough to surround entities belonging to the categories. On the other hand, they need to be tight enough to prevent entities of other categories included.

This can be done in the following steps:

1) Perform contrastive learning so that entities labeled the same group cluster more tightly and different groups separate further.
2) Use high precision data to get the center of each category. This is done using the mean. The initial radius of the boundaries is randomly selected.
3) If an entity is surrounded in category A and actually labeled in category B, we narrow our boundary. Likewise we broaden our boundary if entity is not included in A but labeled A.

with the trained boundaries we could select and update our high decision dataset.

4 Distillation learning



Basically what Bert does is give the words a vector representation.

Since not all entities could be recognized by the rules, a lot of entities remain unlabeled. To make use of these data, we could use the Bert encoder we trained and feed it with unlabeled entities (Since these entities will not enter the Bert encoder due to the rules) and give them psuedo labels. We Then train our model again based on the full dataset.

Another advantage is that our entities is 1-hot labeled to represent the categories, but the psuedo labels are represented by a continuous probability distribution. For Instance Beijing could be labeled [1,0,0] but the Bert encoder could give us [0.7,0.2,0.1]. Such scheme could also potentially increase the effectiveness while training the model.

5 Dataset

We conduct extensive experiments on two different dataset:

1. BC5CDR [2] has 1500 PubMed articles with 15,953 chemical and 13,318 disease entities.
2. CoNLL2003 [3] only uses Person, Location, and Organization entities
   - Training set: 14987 sentences
   - Development (Validation) set: 3466 sentences
   - Testing set: 3684 sentences.

2 J. Li, Y. Sun, R. J. Johnson, D. Sciaky, Z. Lu, Biocreative v cdr task corpus: a resource for chemical disease relation extraction, Database. The Journal of Biological Databases and Curation 2016 (2016)

3 E. Sang, F. D. Meulder, Introduction to the conll-2003 shared task: Language-independent named entity recognition, arXiv (2003).

6 Metrics of Evaluation

We adopt the precision, recall, and macro-averaged F1 scores on two test sets. The precision is used to measure the proportion of correct positive predictions out of all positive predictions:

$$\texttt{Precision} = \frac{\texttt{True Positives}}{\texttt{True Positives} + \texttt{False Positives}} \tag{1}$$

Recall measures the proportion of correct positive predictions out of all the actual positive instances in the dataset:

$$\texttt{Recall} = \frac{\texttt{True Positives}}{\texttt{True Positives} + \texttt{False Negatives}} \tag{2}$$

Macro-averaged F1 is a measure of the overall performance of the model. It is calculated as the harmonic mean of precision and recall, with equal weight given to each metric. Macro-averaged F1 is often used when the dataset is imbalanced, meaning that there are many more instances of one class than another. This metric gives equal importance to all classes, regardless of their size, and can provide a better overall evaluation of the model's performance. The F1 score for a single class is the harmonic mean of precision and recall, given by:

$$\texttt{F1 score} = \frac{2 \times \texttt{Precision} \times \texttt{Recall}}{\texttt{Precision} + \texttt{Recall}} \tag{3}$$

7 Baseline method

We compare the performance of our work with different models

- Seed Rule: a data-driven approach that uses a small set of initial rules or patterns (seeds) to generate a larger set of rules or patterns through iterative refinement and expansion.
- Seed Rule + Neural Tagger: After applying the seed rules, we use the generated weakly labels to train our neural tagger and report the results of the tagger without iterative learning.
- Self-training: We obtain weak labels by applying seed rules, and use weak labels as initial supervision to build a self-training system.

- LinkedHMM [4]: The framework proposes weakly supervised training for sequence labeling using multiple heuristic rules and introduces a new linking rule voting method for grouping sequence elements with the same label.

- HMM-agg [5]: Unlike previous weakly supervised methods, this approach allows the label function to produce probabilistic predictions, which are then combined using a hidden Markov model with parameters estimated by the Baum-Welch algorithm to learn neural sequence marker models.

- CGExpan [6]: The framework introduces an ensemble extension and an automatic class name generation algorithm that uses pretrained language models to produce high-quality class names, guiding the expansion process and addressing semantic drift by filtering the collection iteratively.

- AutoNER [7]: This architecture designs two neural architectures, the FuzzyLSTM-CRF model with the improved IOBES tagging scheme and the new Tie or Break scheme, using only dictionaries to learn efficient NER models.

- TALLOR [8]: this architecture proposes to use high-quality logical rules to train a neural tagger in an automated manner. It includes the use of compound rules to increase the precision of boundary detection and generate more diverse pseudo labels. Meanwhile, it also introduces a dynamic label selection strategy to ensure the quality of pseudo labels and prevent overfitting.

4 E. Safranchik, S. Luo, S. Bach, Weakly supervised sequence tagging from noisy rules, in Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 5570–5578.

5 P. Lison, A. Hubin, J. Barnes, S. Touileb, Named entity recognition without labelled data: A weak supervision approach, arXiv preprint arXiv:2004.14723 (2020).

6 Y. Zhang, J. Shen, J. Shang, J. Han, Empower entity set expansion via language model probing, arXiv preprint arXiv:2004.13897 (2020)

7 J. Shang, L. Liu, X. Gu, X. Ren, T. Ren, J. Han, Learning named entity tagger using domain-specific dictionary, in: 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018, Association for Computational Linguistics, 2020, pp. 2054–2064.

8 J. Li, H. Ding, J. Shang, J. McAuley, Z. Feng, Weakly supervised named entity tagging with learnable logical rules, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 4568–4581.

Unveiling Causal Structures in Biological Data

CS573 Course Project

Shyaman Jayasundara
Jasorsi Ghosh
04/16/2024

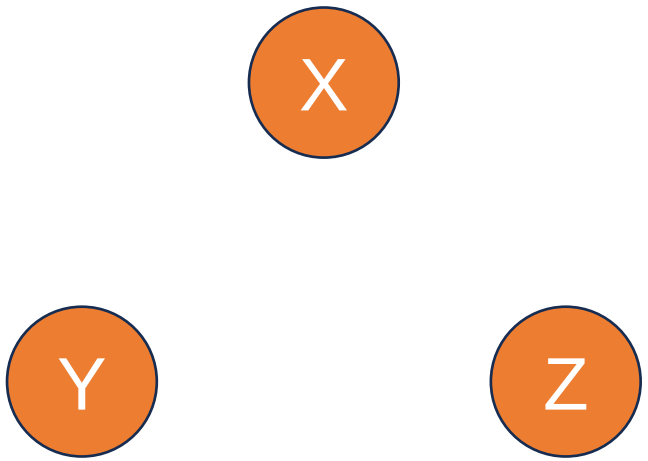
Correlation vs Causation

Correlation vs Causation

Correlation - statistical association

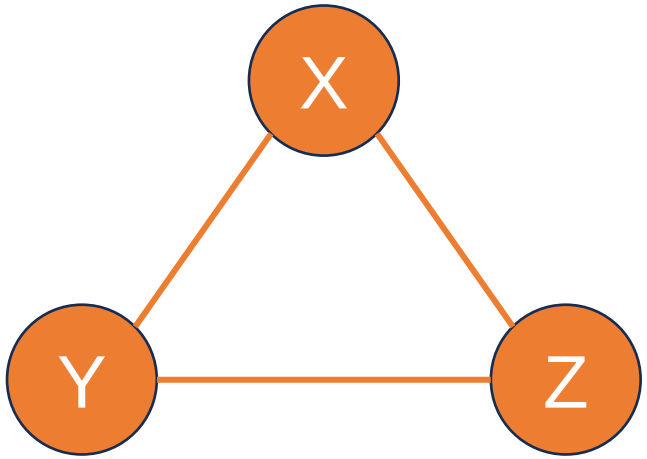
Correlation vs Causation

Correlation - statistical association



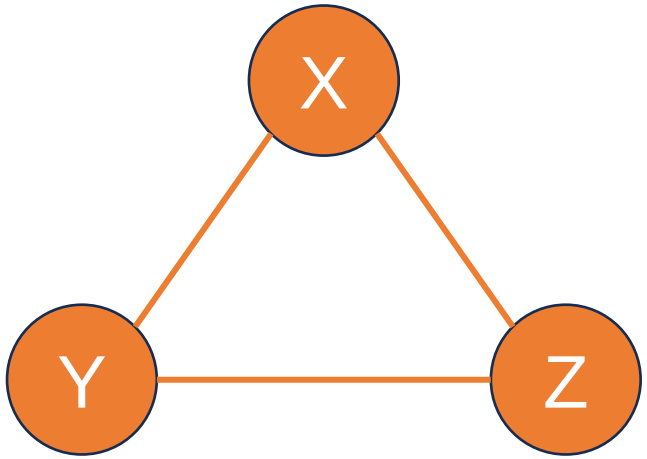
Correlation vs Causation

Correlation - statistical association

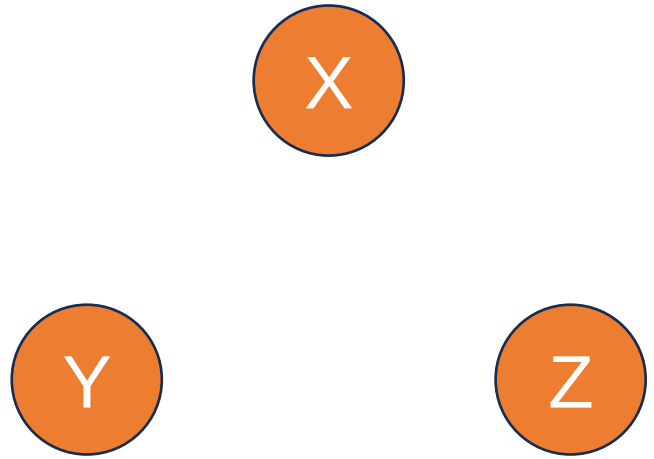


Correlation vs Causation

Correlation - statistical association

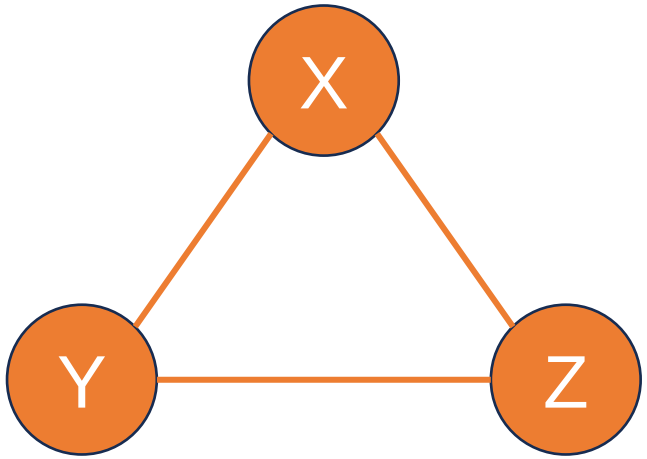


Causation - cause-and-effect

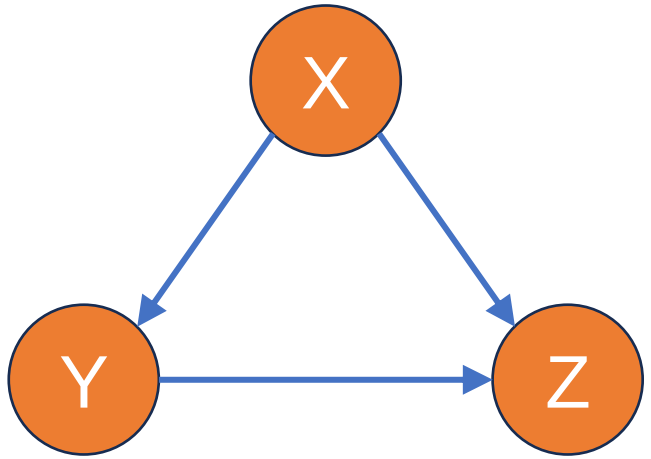


Correlation vs Causation

Correlation - statistical association

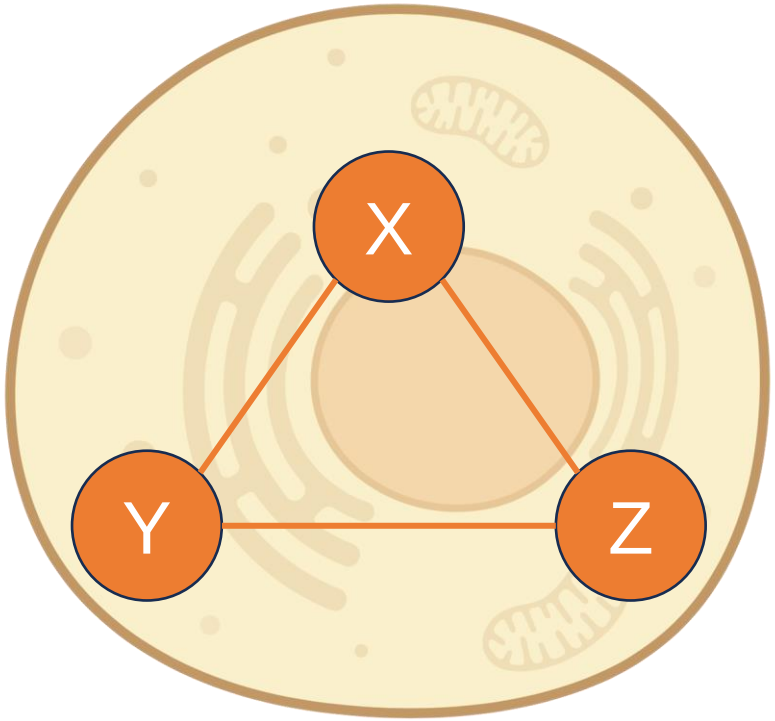


Causation - cause-and-effect

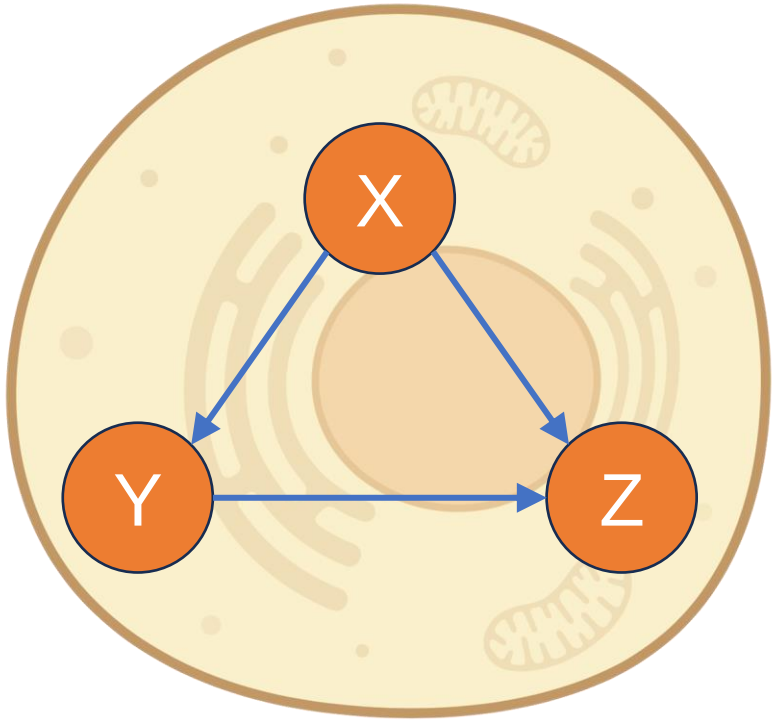


Correlation vs Causation

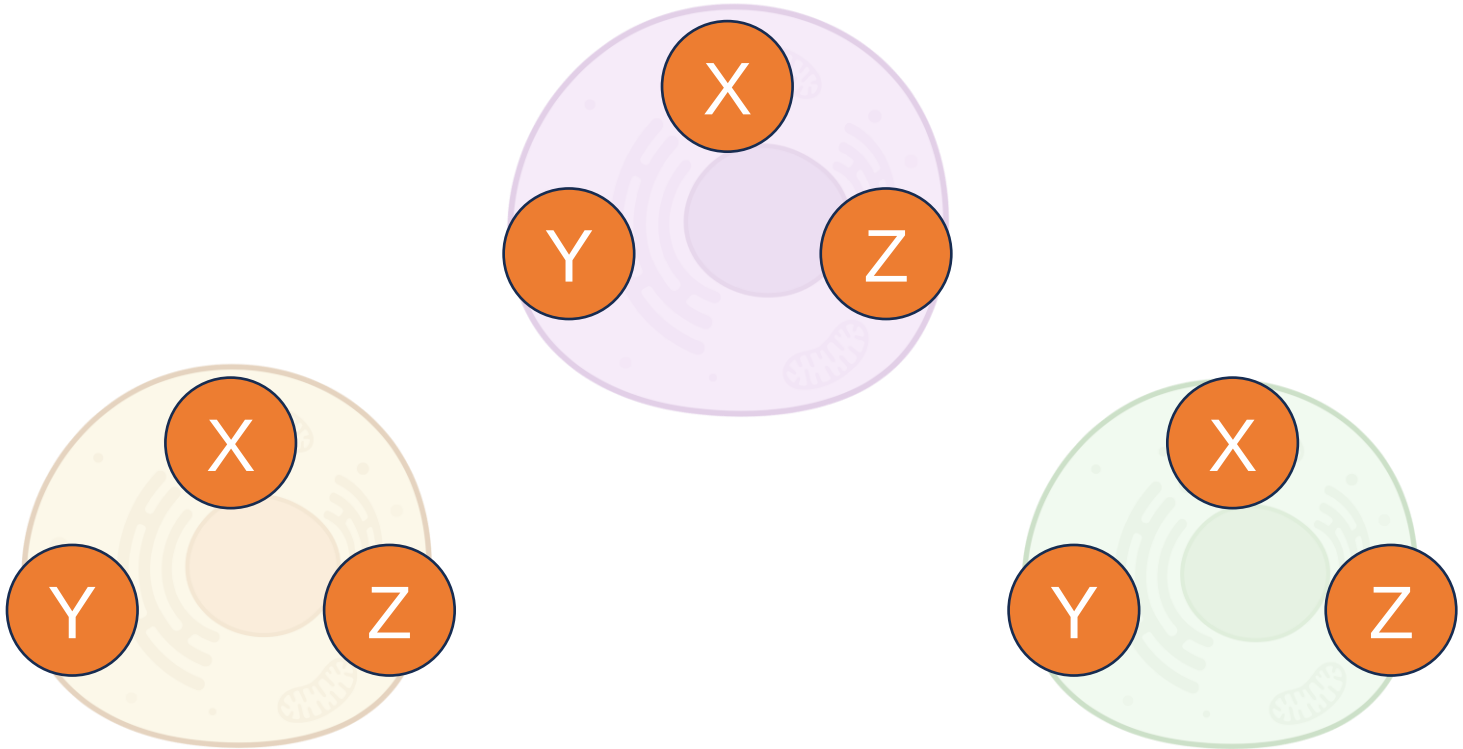
Correlation - statistical association



Causation - cause-and-effect



Interventional data to identify causal structures

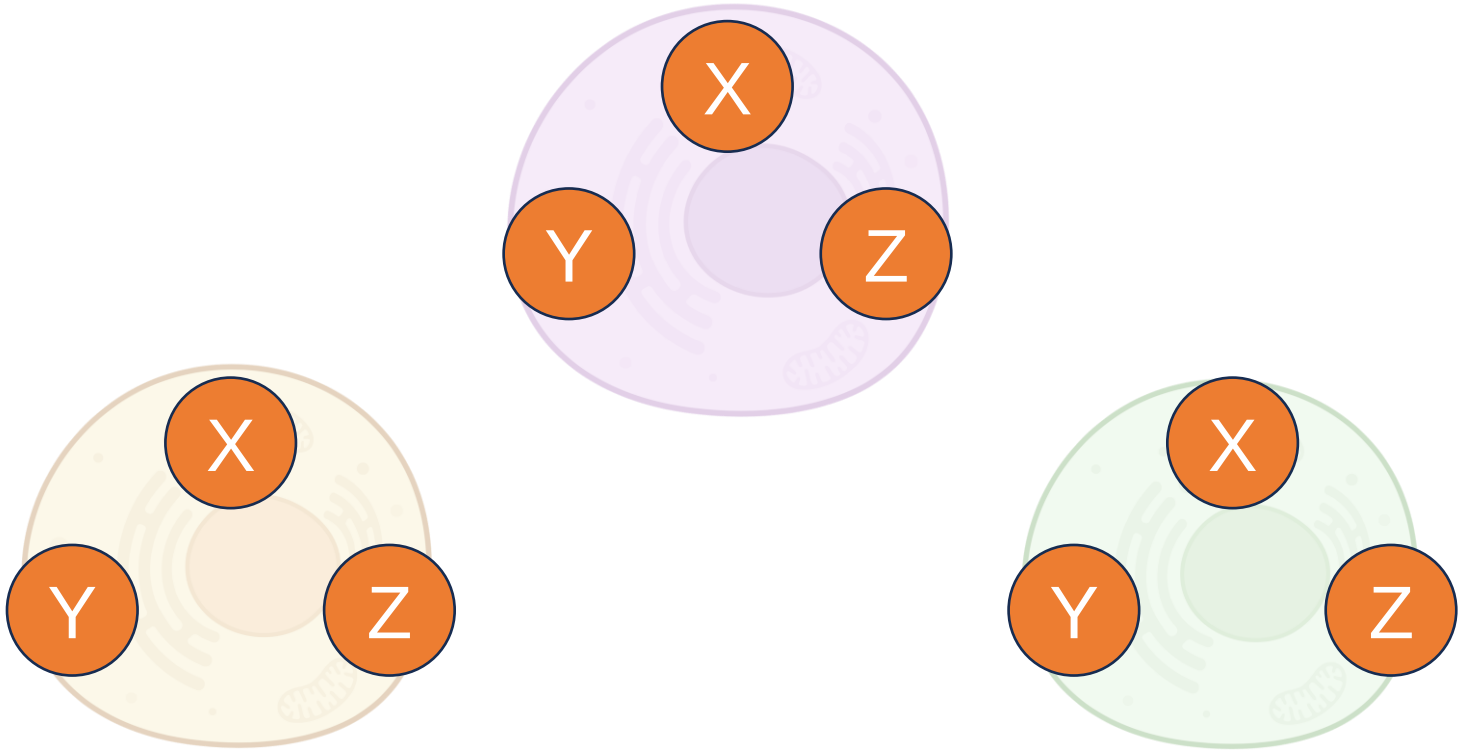


Interventional data to identify causal structures

Observation data: expression of genes using RNA sequencing

+

Intervention: gene editing technology

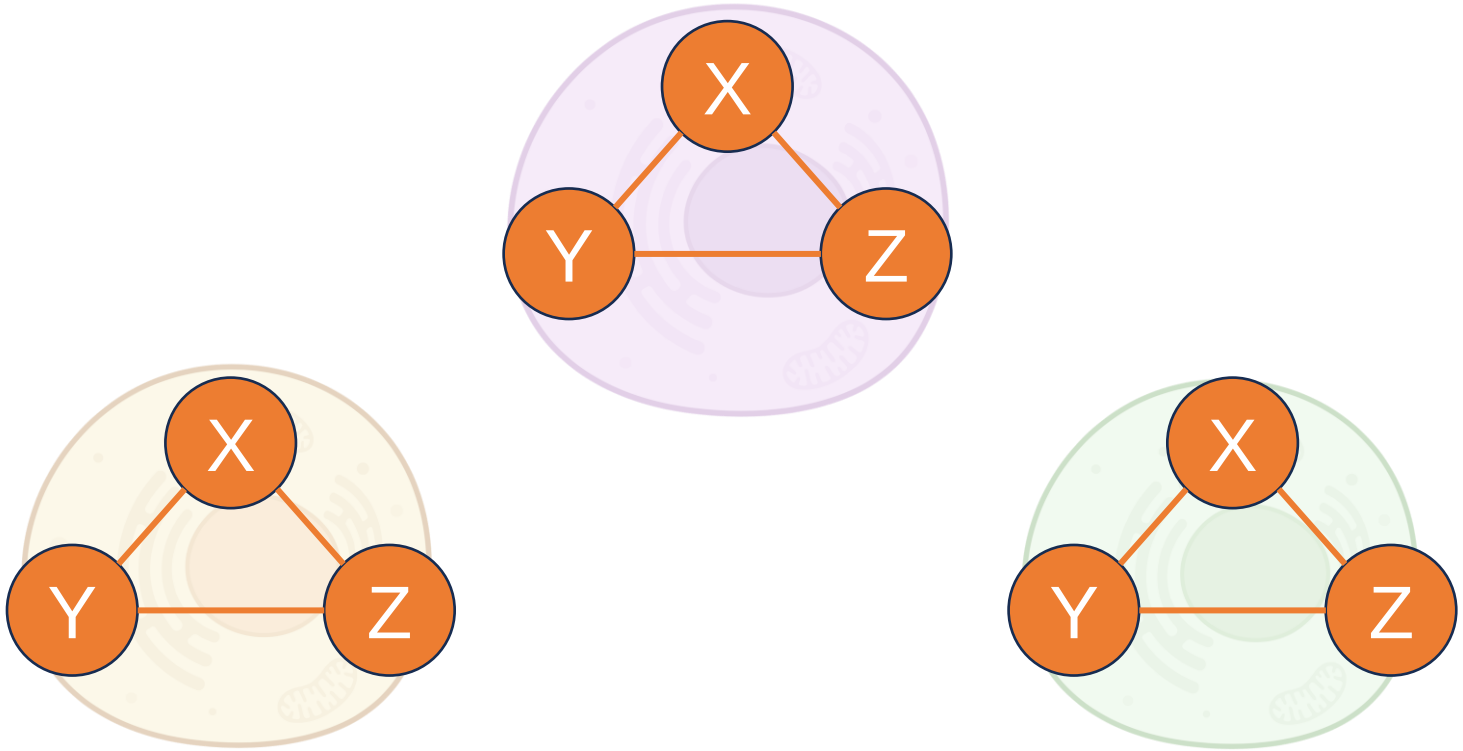


Interventional data to identify causal structures

Observation data: expression of genes using RNA sequencing

+

Intervention: gene editing technology

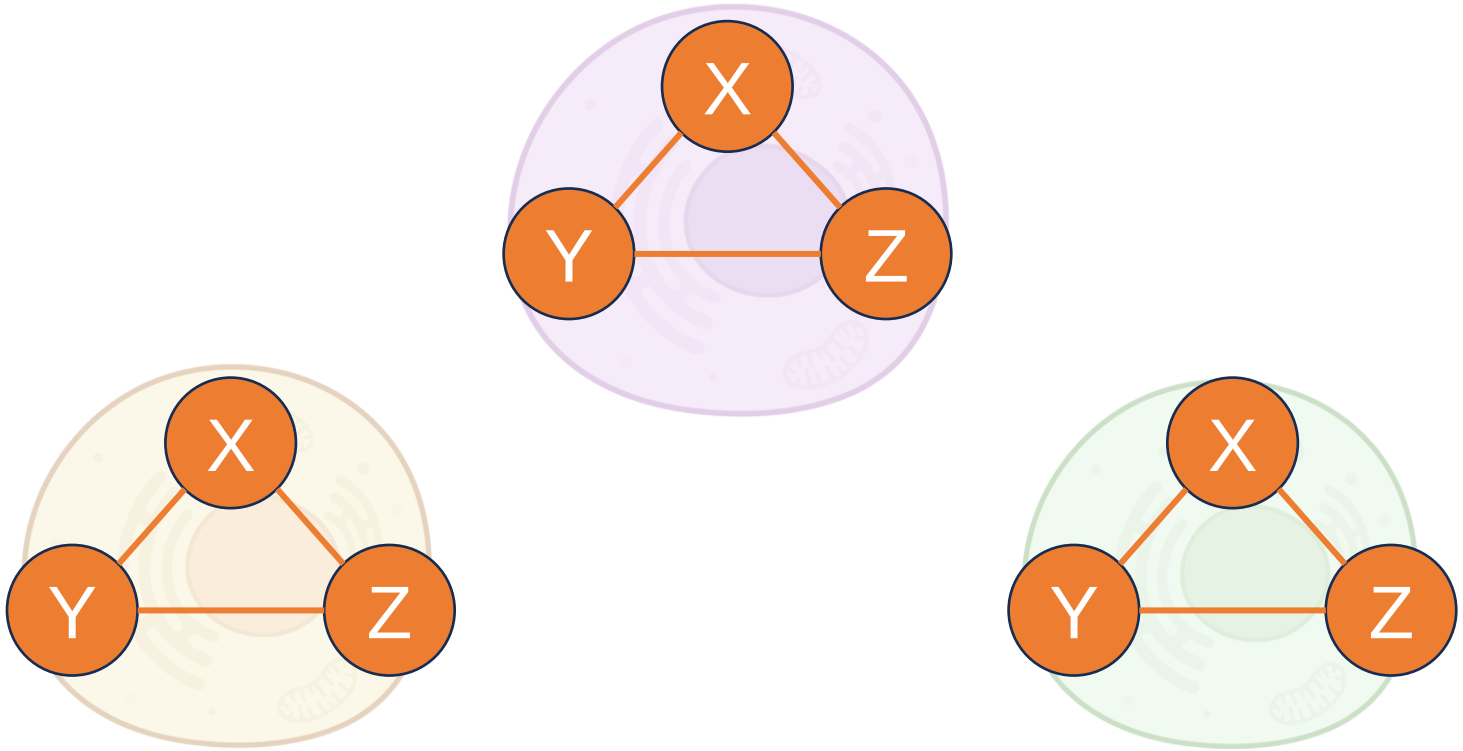


Interventional data to identify causal structures

Observation data: expression of genes using RNA sequencing

+

Intervention: gene editing technology

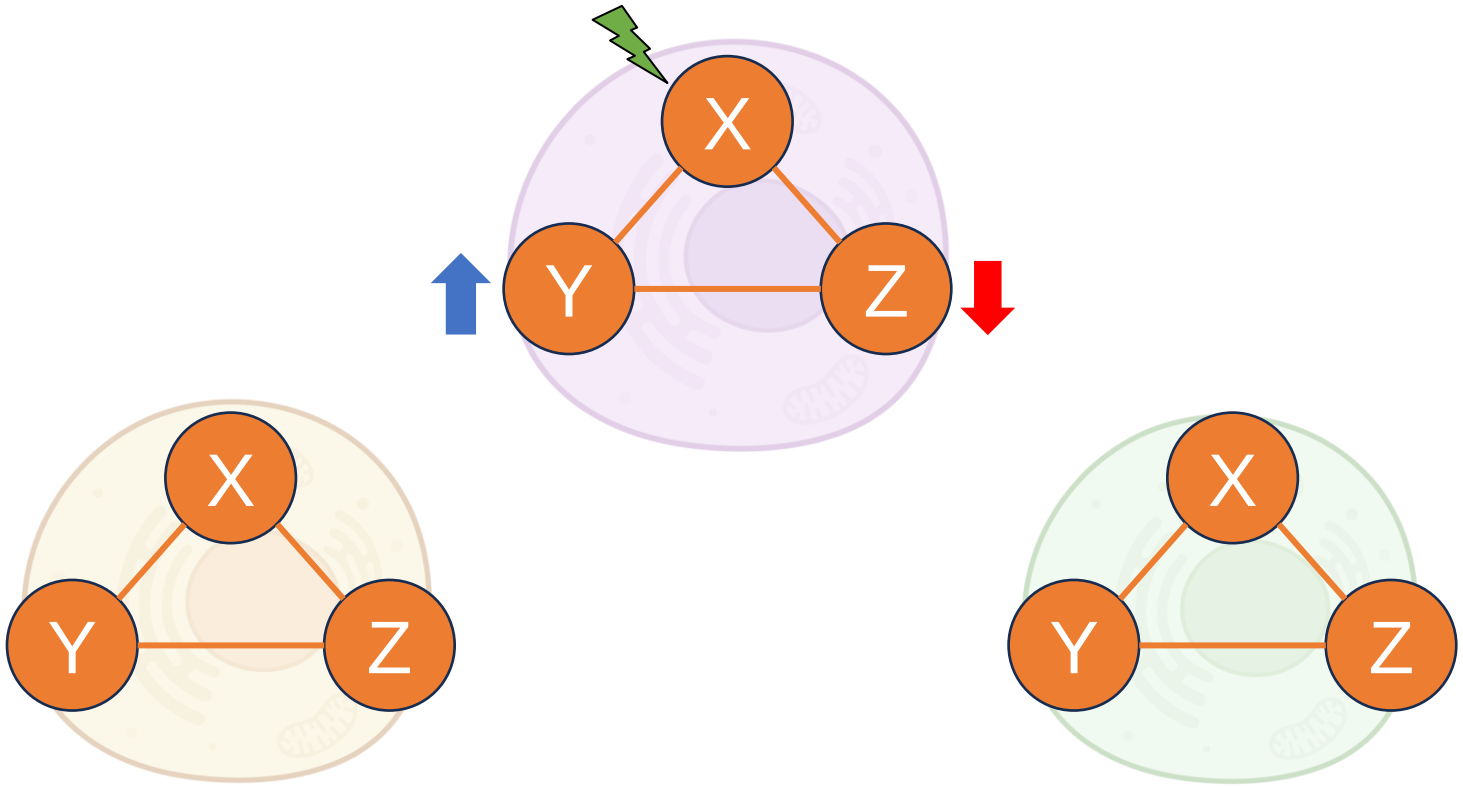


Interventional data to identify causal structures

Observation data: expression of genes using RNA sequencing

+

Intervention: gene editing technology

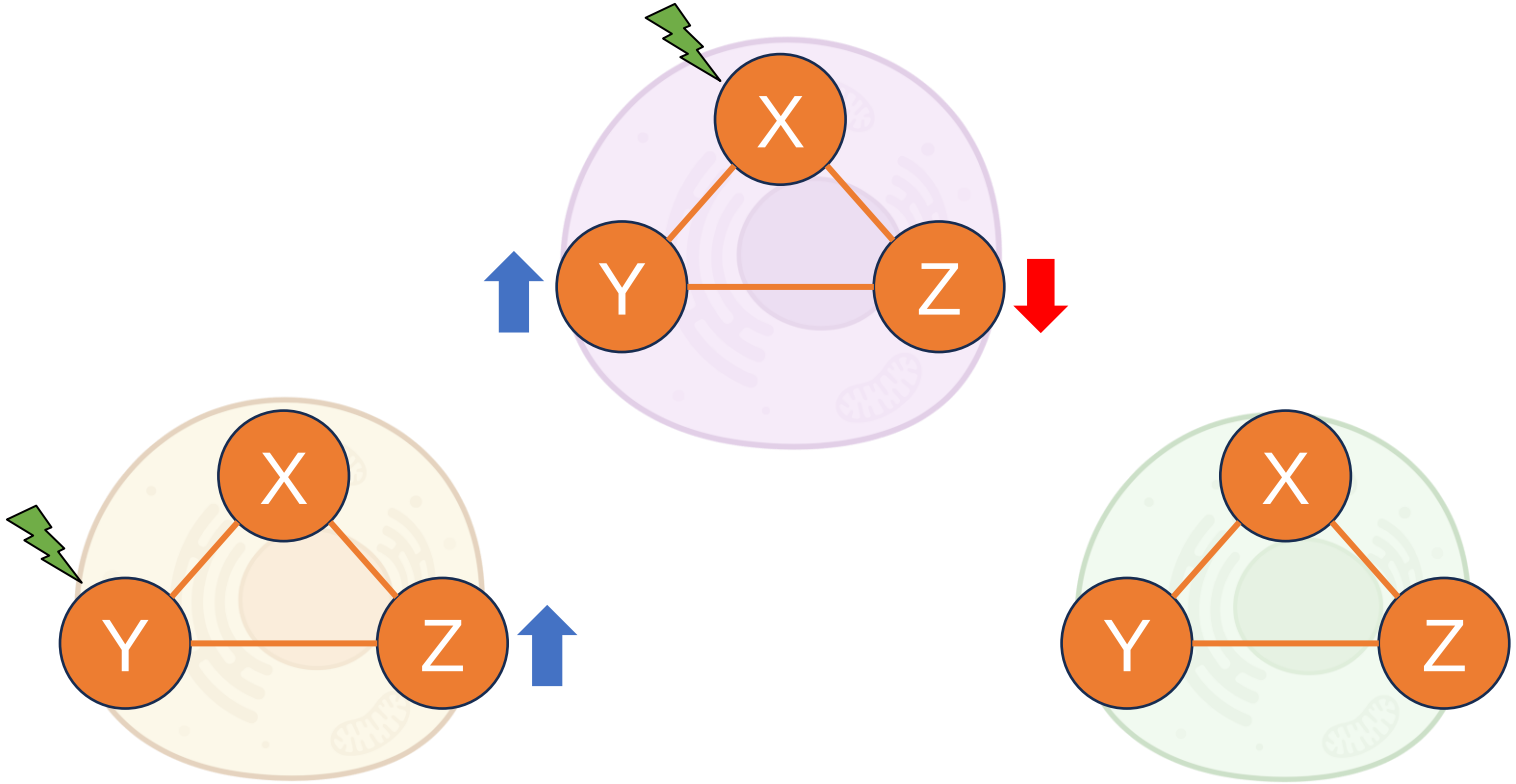


Interventional data to identify causal structures

Observation data: expression of genes using RNA sequencing

+

Intervention: gene editing technology

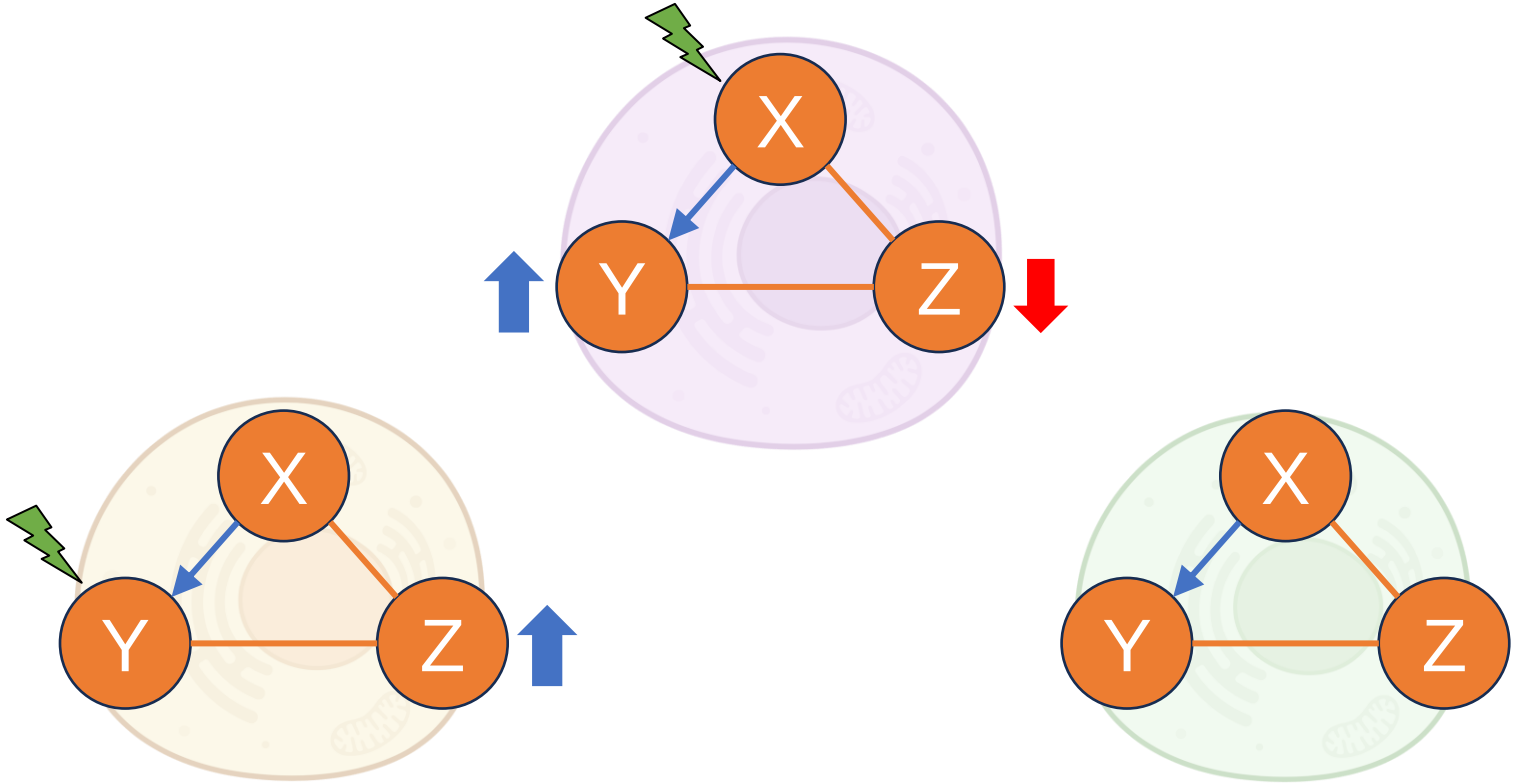


Interventional data to identify causal structures

Observation data: expression of genes using RNA sequencing

+

Intervention: gene editing technology

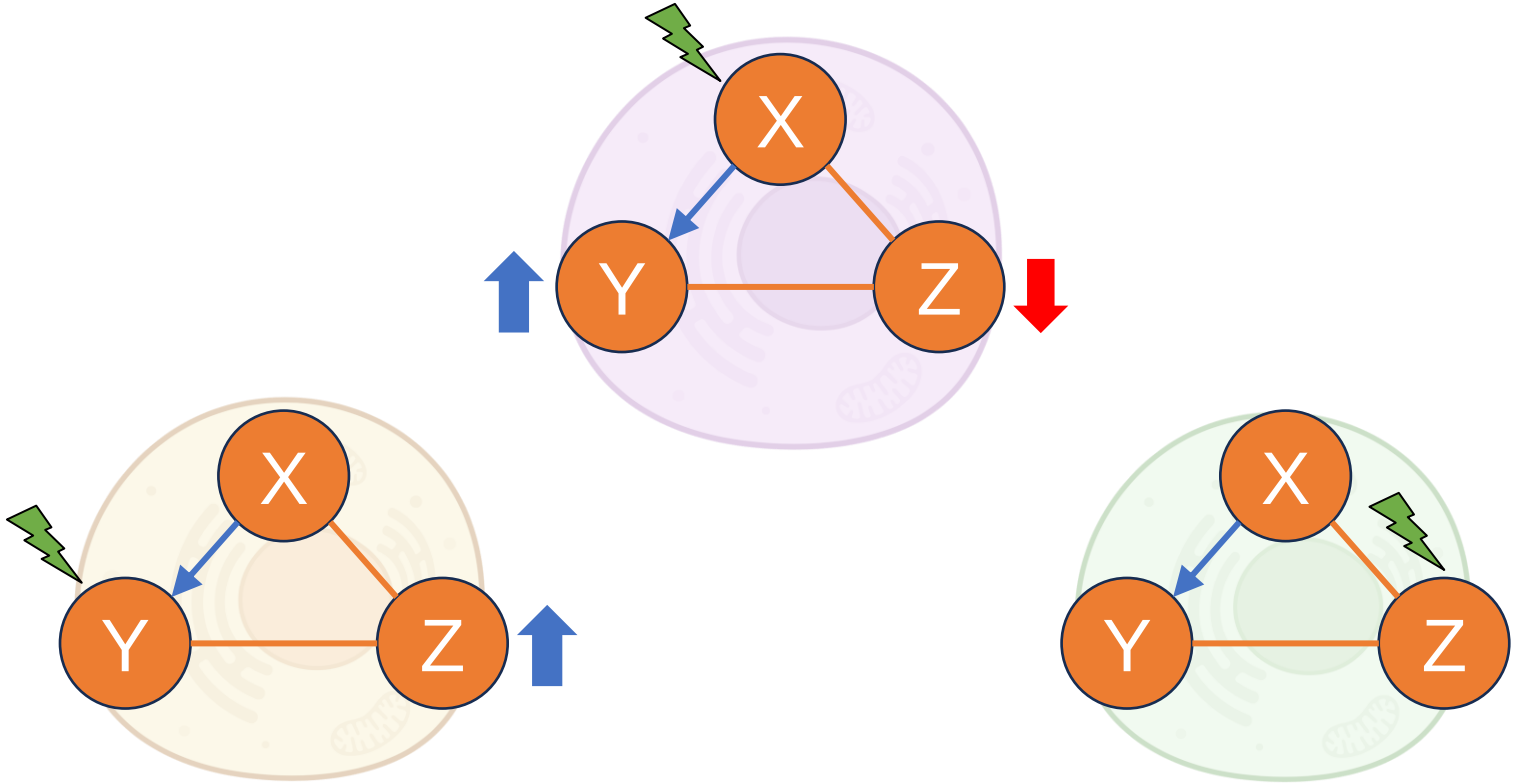


Interventional data to identify causal structures

Observation data: expression of genes using RNA sequencing

+

Intervention: gene editing technology

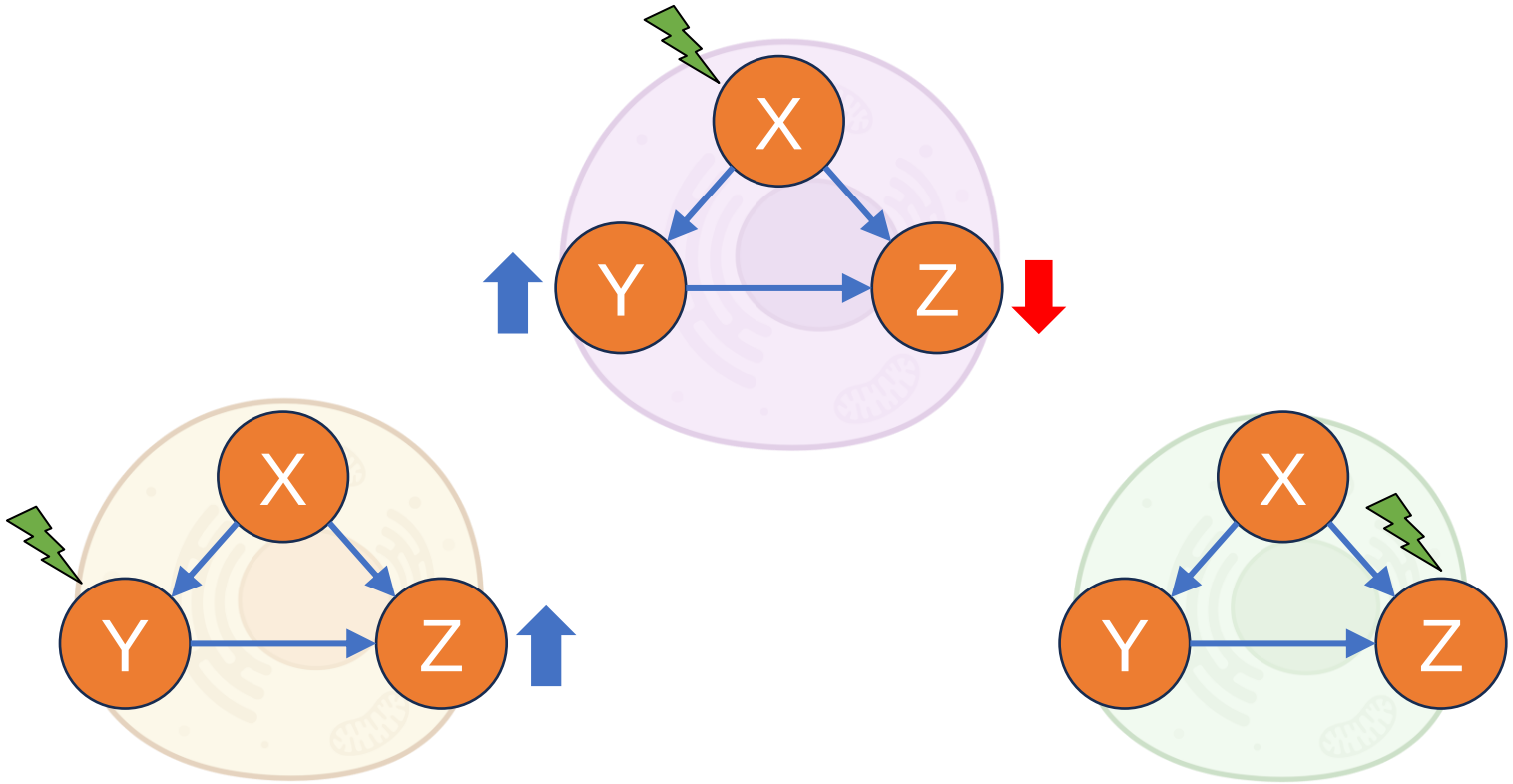


Interventional data to identify causal structures

Observation data: expression of genes using RNA sequencing

+

Intervention: gene editing technology

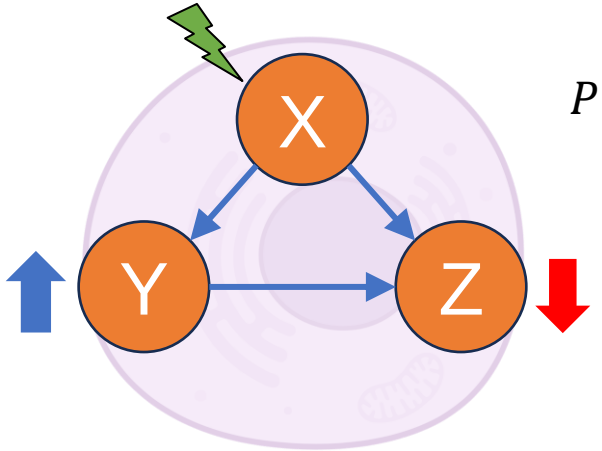


Interventional data to identify causal structures

Observation data: expression of genes using RNA sequencing

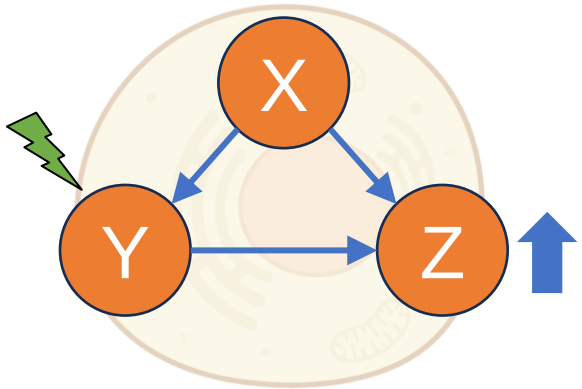
+

Intervention: gene editing technology

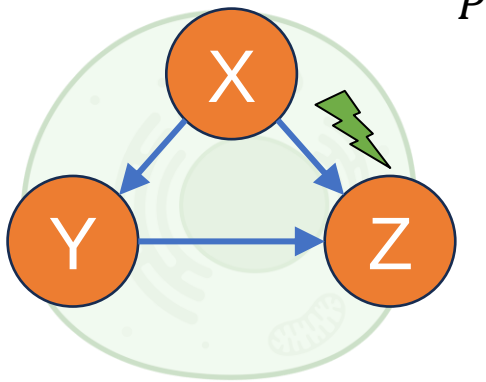


$$P(Y|X,Z) = P(Y|\, do(X), Z)$$

$$P(X|do(Y)) = P(X)$$



$$P(X|do(Z)) = P(X)$$

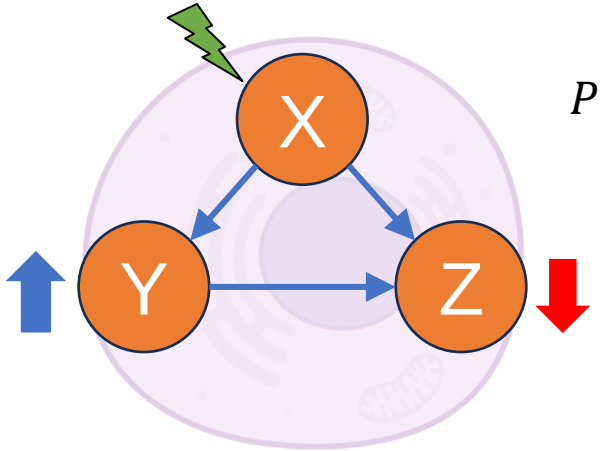


Interventional data to identify causal structures

Observation data: expression of genes using RNA sequencing

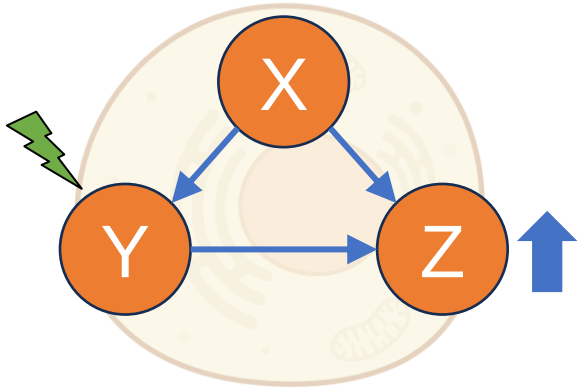
+

Intervention: gene editing technology

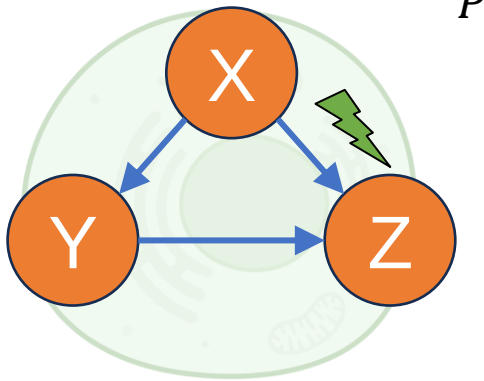


$P(Y|X,Z) = P(Y|\, do(X),Z)$

$P(X|do(Y)) = P(X)$



$P(X|do(Z)) = P(X)$



Task: DAG (Directed Acyclic Graph) learning

Interventional data to identify causal structures

Observation data: expression of genes using RNA sequencing

+

Intervention: gene editing technology

$P(Y|X,Z) = P(Y|\,do(X),Z)$

$P(X|do(Z)) = P(X)$

- Dataset of 90,000 cells (samples)
- 89 different gene mutations (interventions) to TP53 gene
- Expression measured for 100 genes (observed variables)

Ursu et. al, 2022

Task: DAG (Directed Acyclic Graph) learning

Interventional data to identify causal structures

Observation data: expression of genes using RNA sequencing

+

Intervention: gene editing technology

$P(Y|X,Z) = P(Y|\,do(X),Z)$

$P(X|do(Z)) = P(X)$

- Dataset of 90,000 cells (samples)
- 89 different gene mutations (interventions) to TP53 gene
- Expression measured for 100 genes (observed variables)

Ursu et. al, 2022

Task: DAG (Directed Acyclic Graph) learning

Interventional data to identify causal structures

Observation data: expression of genes using RNA sequencing

+

Intervention: gene editing technology

$P(Y|X,Z) = P(Y|\,do(X),Z)$

$P(X|do(Z)) = P(X)$

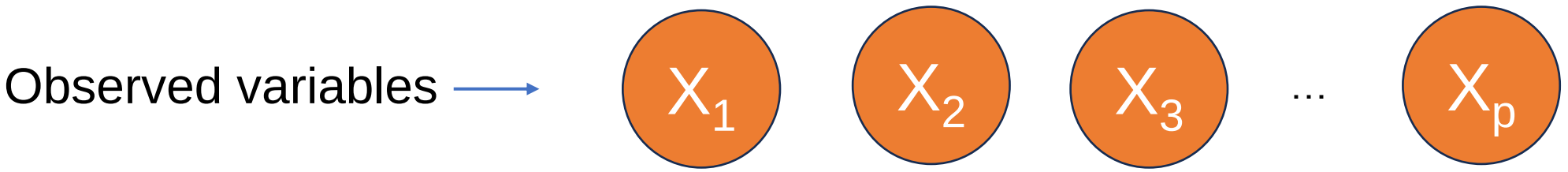
- Dataset of 90,000 cells (samples)
- 89 different gene mutations (interventions) to TP53 gene
- Expression measured for 100 genes (observed variables)

Ursu et. al, 2022

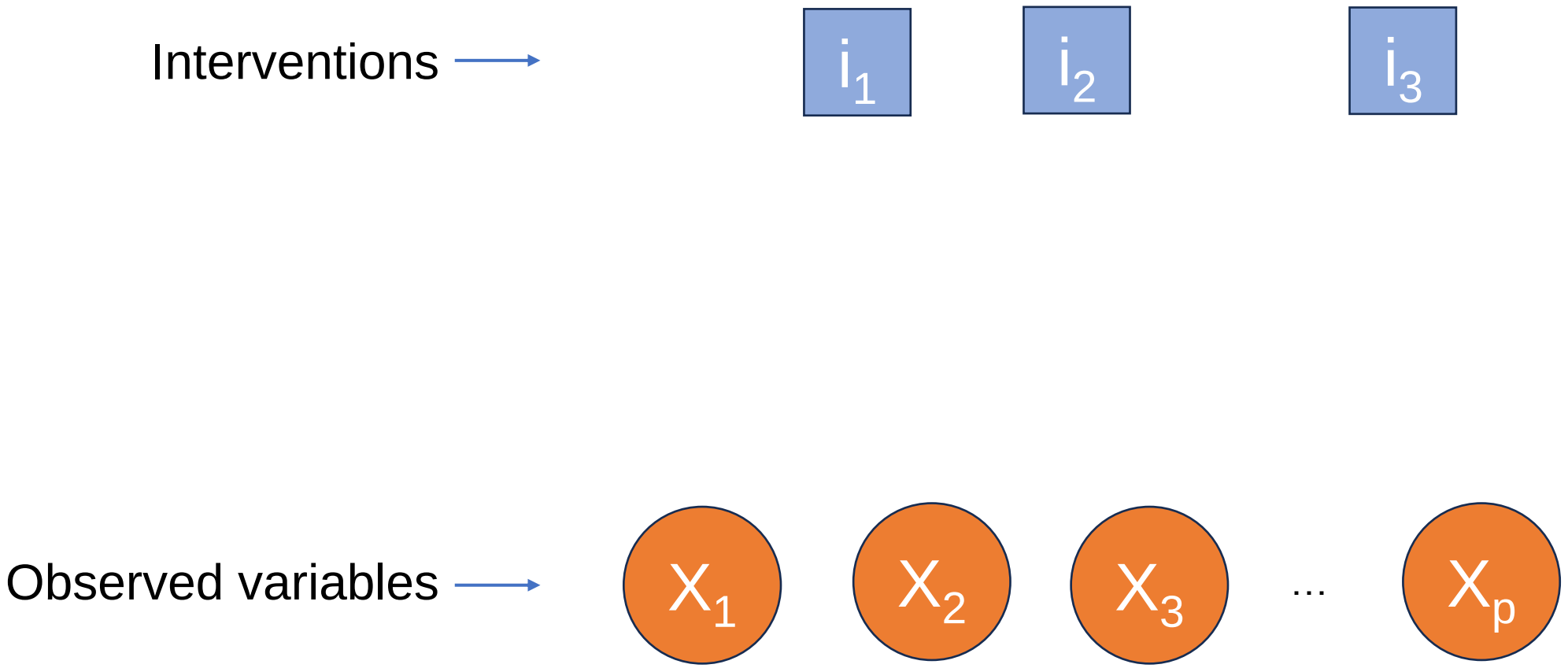
Task: DAG (Directed Acyclic Graph) learning

Observed variables are generated in two steps

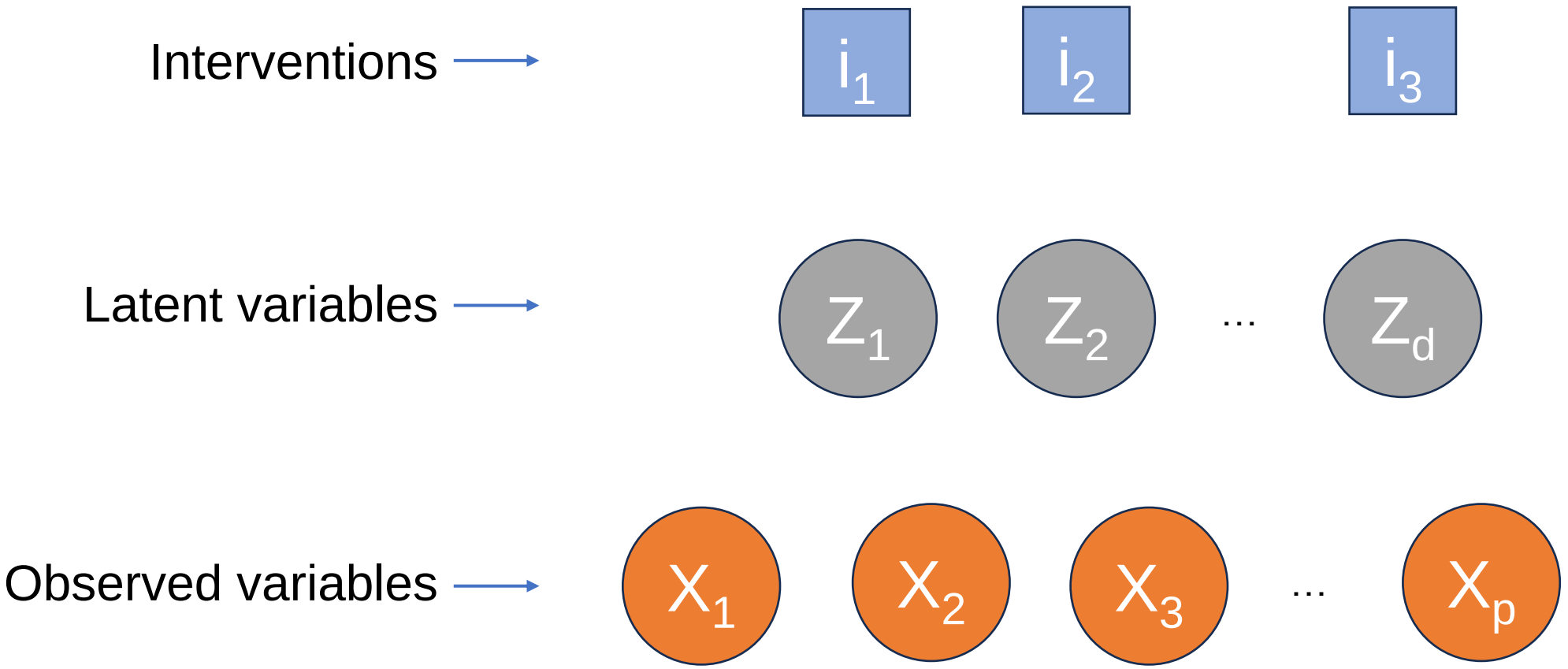
Observed variables are generated in two steps



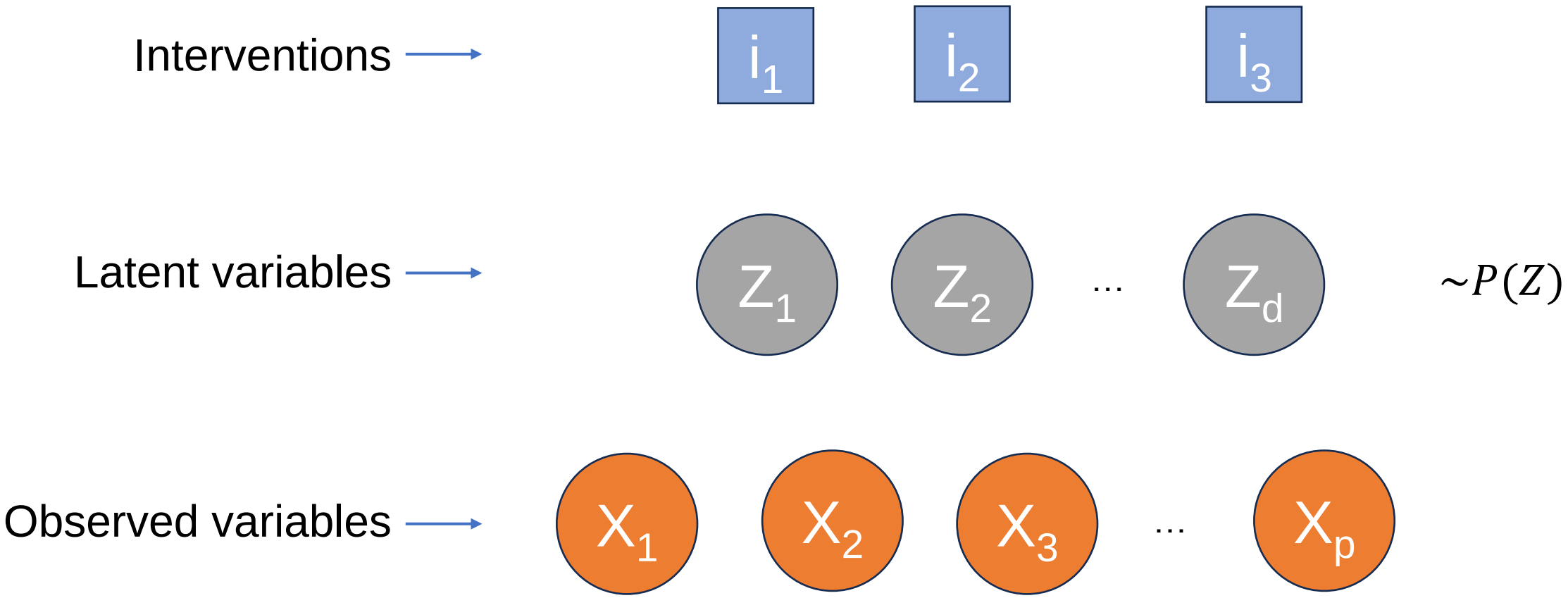
Observed variables are generated in two steps



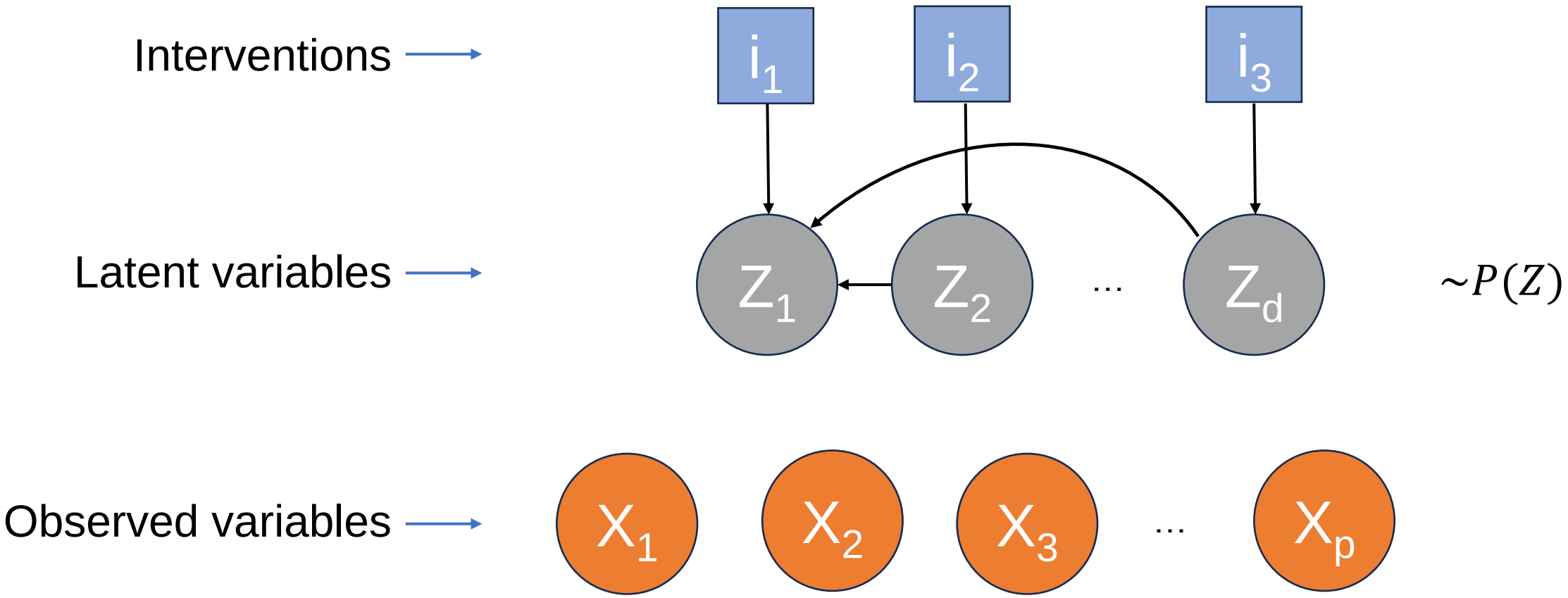
Observed variables are generated in two steps



Observed variables are generated in two steps



Observed variables are generated in two steps

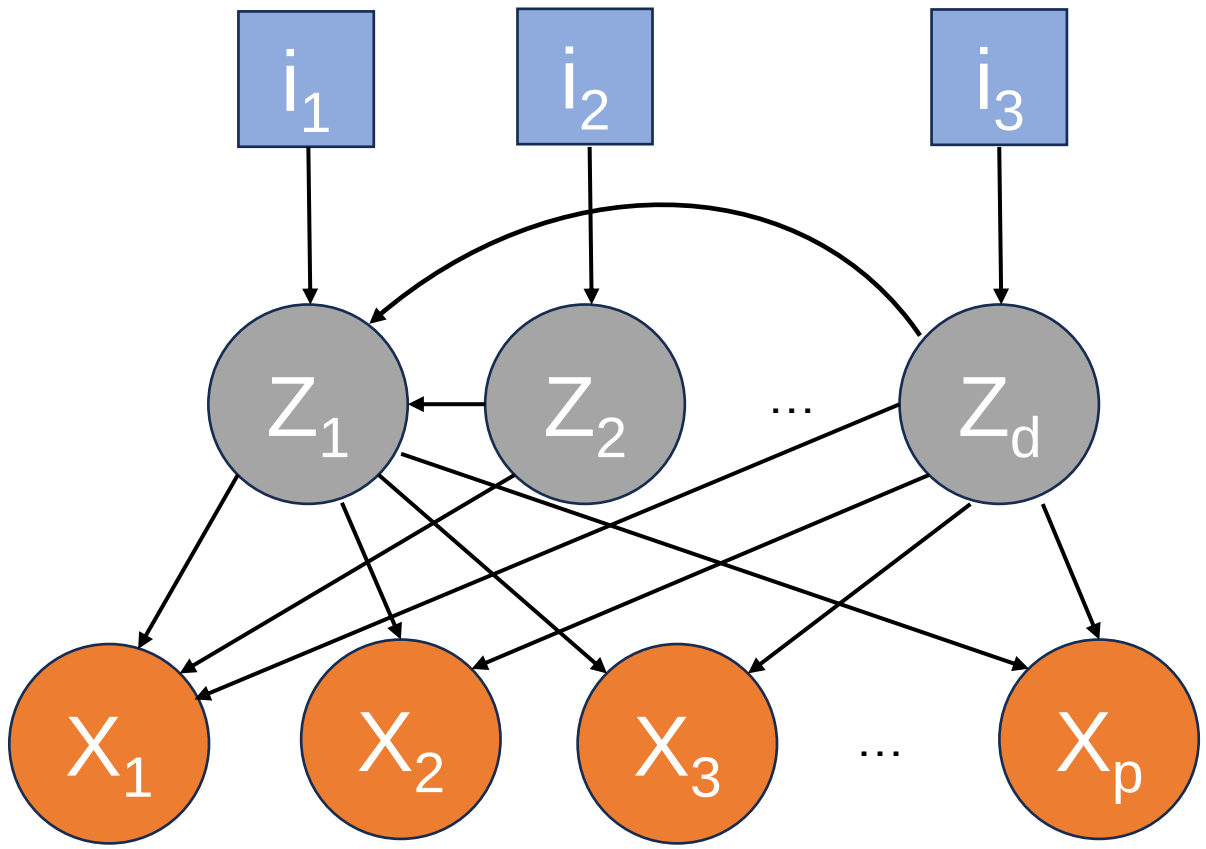


Observed variables are generated in two steps

Interventions →

Latent variables →

Observed variables →



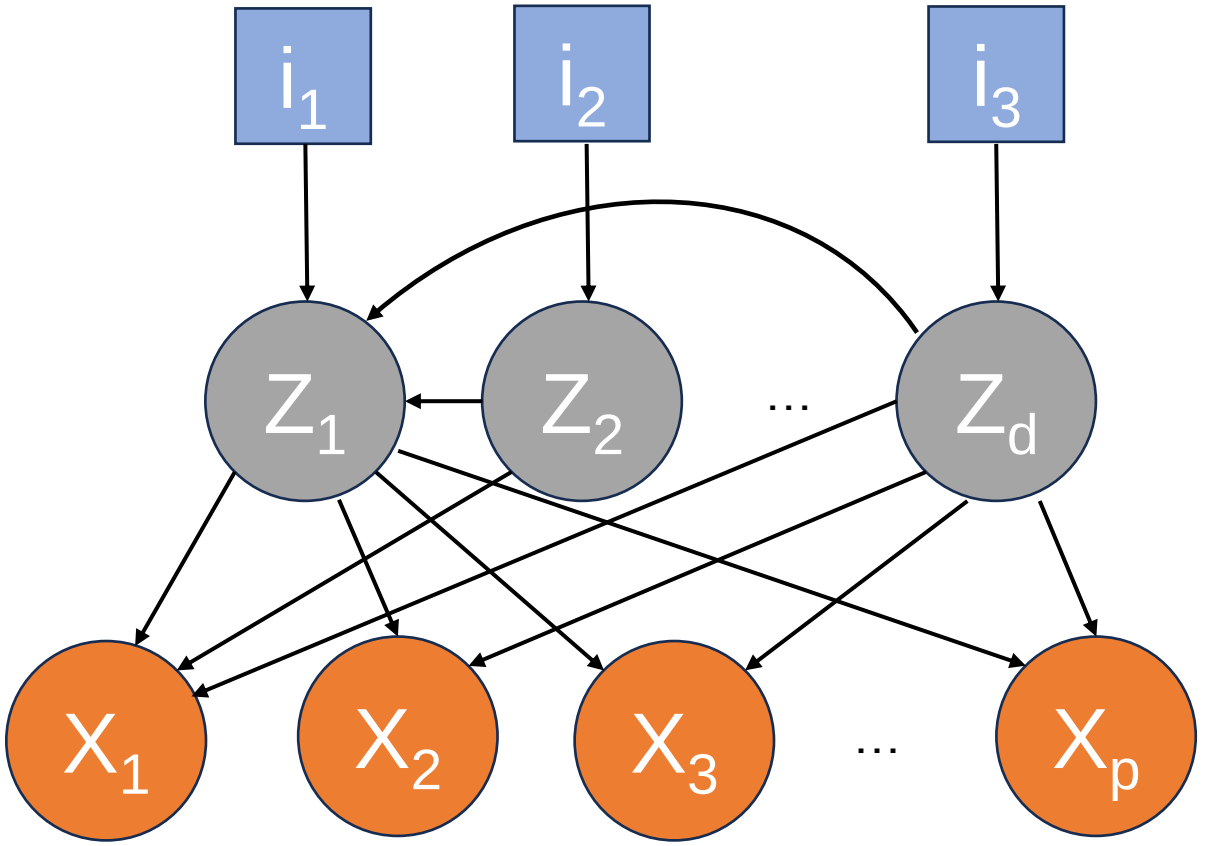
$\sim P(Z)$

Observed variables are generated in two steps

Interventions →

Latent variables →

Observed variables →



$\sim P(Z)$

$X = G(Z)$

G - Injective linear function/ Mixing function

Approaches for Causal disentanglement

Approaches for Causal disentanglement

Data  A unique model
that generated the data

Approaches for Causal disentanglement

Data  A unique model that generated the data

1. Restriction on the latent DAG

> ICA (Comon,1994)

Approaches for Causal disentanglement

Data  A unique model that generated the data

1. Restriction on the latent DAG

> ICA (Comon,1994)

2. Restriction on the mixing function

> Causal discovery for mixed linear data (Yan, 2021, Xie 2022)

Approaches for Causal disentanglement

Data  A unique model that generated the data

1. Restriction on the latent DAG
> ICA (Comon,1994)

2. Restriction on the mixing function
> Causal discovery for mixed linear data (Yan, 2021, Xie 2022)

3. Learning from different contexts
>Invertible Latent Causal Models (Sean, 2023)

Approaches for Causal disentanglement

Data  A unique model that generated the data

1. Restriction on the latent DAG
> ICA (Comon,1994)

2. Restriction on the mixing function
> Causal discovery for mixed linear data (Yan, 2021, Xie 2022)

3. Learning from different contexts
>Invertible Latent Causal Models (Sean, 2023)

Approaches for Causal disentanglement

Data → A unique model that generated the data

1. Restriction on the latent DAG
> ICA (Comon,1994)

2. Restriction on the mixing function
> Causal discovery for mixed linear data (Yan, 2021, Xie 2022)

3. Learning from different contexts
>Invertible Latent Causal Models (Sean, 2023)

Assumptions:
1. Linear Mixing functions: $X = G(Z)$, G has full column rank mixing function.
2. Linear latent Models: The SCM is linear.
3. Single Node Intervention

Sufficient and necessary conditions for DAG recovery

Sufficient and necessary conditions for DAG recovery

- **Under Perfect Intervention(s):**
  1. One intervention per latent node is sufficient **AND**
  2. in worst case necessary for identification of the model in [K] contexts.

Sufficient and necessary conditions for DAG recovery

- **Under Perfect Intervention(s):**
  1. One intervention per latent node is sufficient **AND**
  2. in worst case necessary for identification of the model in [K] contexts.

Constructive approach: $\theta_k - \theta_0 = (H^T B_k^T e_{i_k})^{\otimes 2} - (H^T B_0^T e_{i_k})^{\otimes 2}$

θ_i: Precision matrix in context i.
H: Inverse mapping function
B_k: Intervention matrix in context k
e_{i_k}: Choice of intervention variable in context k.
$\otimes$: is the outer-product.

Iteratively recover the DAG

- Recover the source nodes (corresponding row of H)
- Build the partial ordering of nodes over intervention targets.

Iteratively recover the DAG

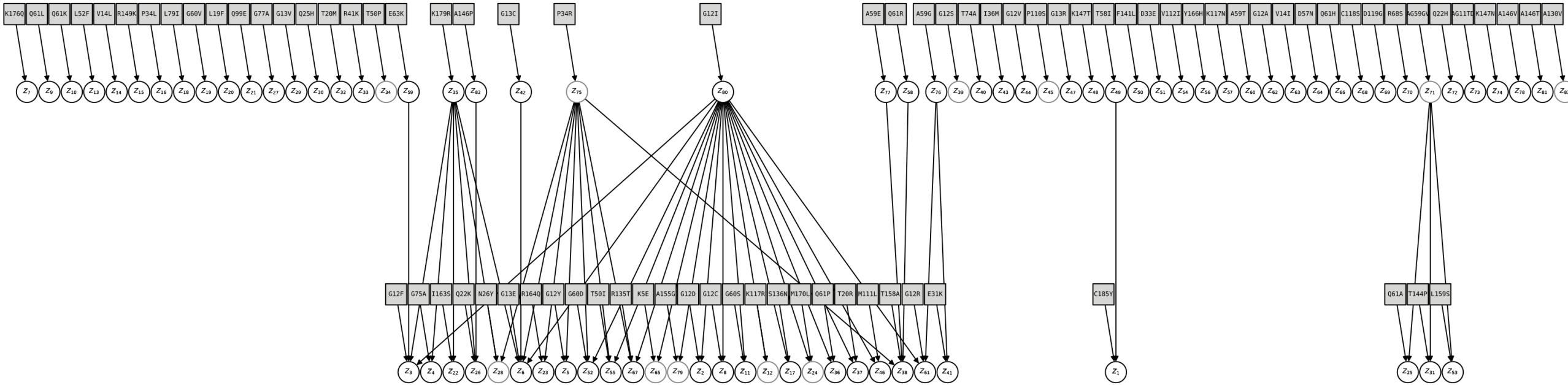
- Recover the source nodes (corresponding row of H)
- Build the partial ordering of nodes over intervention targets.

General Idea: for I being nodes of the DAG

$$rowspan(\theta_k - \theta_0) \subseteq < h_i : i \in I > \iff parents(i_k) \subseteq I$$

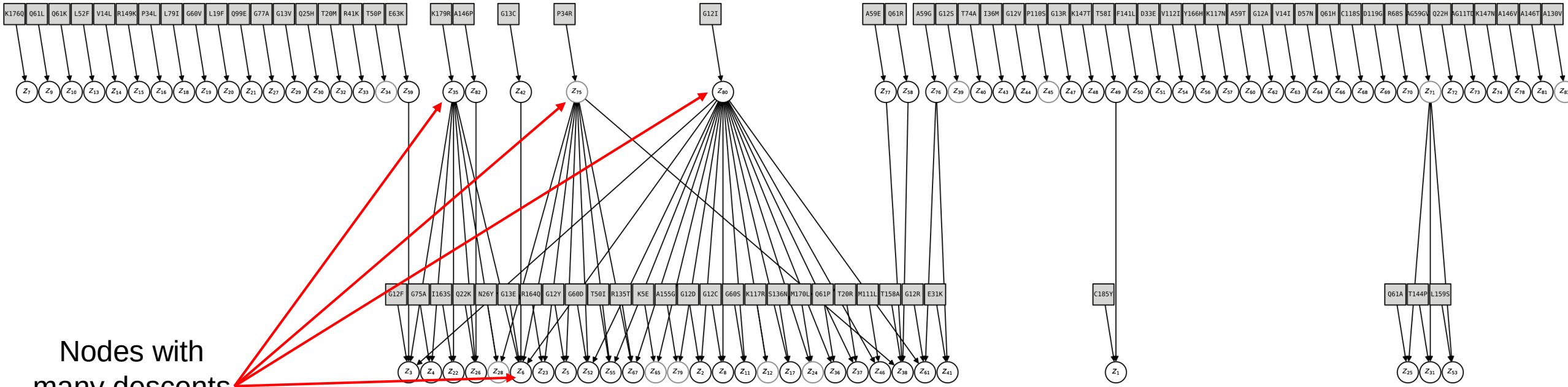
Latent variables are shared by multiple descendants

Latent variables are shared by multiple descendants



The latent graph and intervention targets learned from TP53 mutations

Latent variables are shared by multiple descendants



Nodes with many descents or ancestors

The latent graph and intervention targets learned from TP53 mutations

Future works

- Relaxing the strong assumptions used in the problem formulation.
- Introduce multi-node intervention and recover the graph.
- Use non-linear causal settings.
- Perform statistical analysis of causal disentanglement on the setup.

Thank you!