Incorporating Causal Graphical Prior Knowledge into Predictive Modeling via Simple Data Augmentation

Takeshi Teshima, Masashi Sugiyama

The University of Tokyo, RIKEN

Presentors: Tong Zhang, Zhongqi Yang

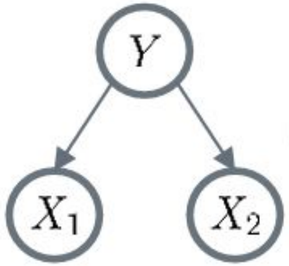
Agenda

- Background & Problem Definition
- General Idea of ADMGs Data Augmentation
- Practical Algorithm
- Results on Real World Data

Causal Graphs (CGs) (Pearl, 2009)

Representation of our knowledge of data generating processes.

CGs imply conditional independence (CI) relations (Pearl, 2009) (Richardson, 2003).

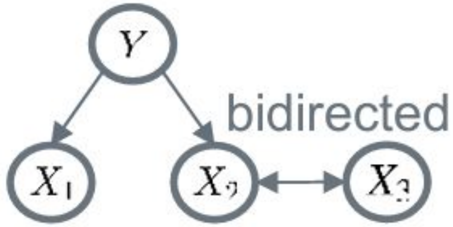


$\Rightarrow X_1 \perp\!\!\!\perp X_2 \mid Y$

Acyclic Directed Mixed Graphs (ADMGs) (Richardson, 2003) (Richardson *et al.*, 2017)

Directed acyclic graphs (possibly) with bidirected edges. $\mathcal{G} = ([D], \mathcal{E}, \mathcal{B})$

Used for causal models with latent variables
(semi-Markov models; cf. Latent projection (Tian *et al.*, 2002)).

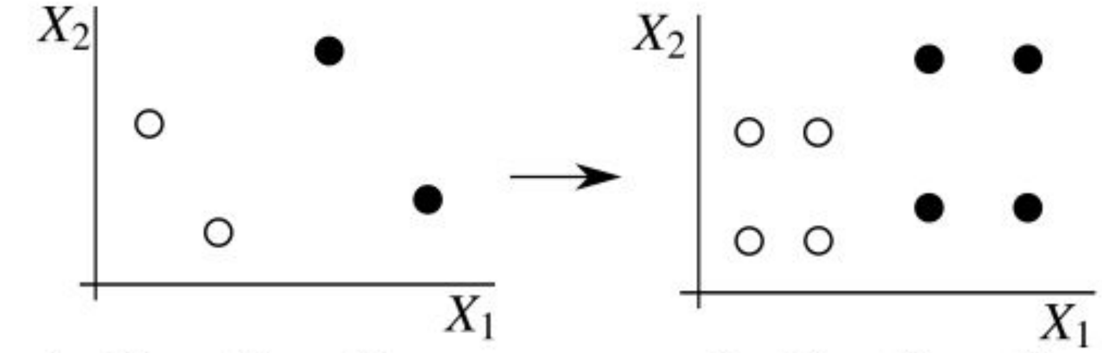


Topological ADMG Factorization (Tian *et al.*, 2002) (Bhattacharya *et al.*, 2020)

Given a semi-Markov model, $p(\mathbf{Z}) = \prod_{j=1}^{D} p_{j|\mathfrak{mp}(j)}(Z^j | \mathbf{Z}^{\mathfrak{mp}(j)})$ holds.

$\mathfrak{mp}(j)$: "Markov pillow" of variable Z^j (Generalization of "parents" in ADMGs.)

Augmentation with a CG



$X_1 \leftarrow Y \rightarrow X_2$

Y	X_1	X_2
○	a	c
○	b	d
●	α	γ
●	β	δ

Y	X_1	X_2
○	a	c
○	a	d
○	b	c
○	b	d

Y	X_1	X_2
●	α	γ
●	α	δ
●	β	γ
●	β	δ

Problem Definition

$\mathbf{Z} = (Z^1, \ldots, Z^D) \sim p$: joint data of X and Y.

(each Z^j may be continuous or discrete)

Main Assumption

- $p(\mathbf{Z})$ satisfies the topological ADMG factorization w.r.t. $\mathcal{G}$

(Bhattacharya *et al.*, 2020)

We are given:

- Labeled data $\mathcal{D} = \{\mathbf{Z}_i\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} p$.
- Estimator $\hat{\mathcal{G}}$ of the underlying ADMG $\mathcal{G}$.

Goal

Find a predictor $f : X \mapsto Y$ with small $R(f) = \mathbb{E}[\ell(f, \mathbf{Z})]$.

$$R(f) = \int_{\mathcal{Z}} \ell(f, \mathbf{Z}) \prod_{j=1}^{D} \underbrace{p_{j|\mathrm{mp}(j)}(Z^j|\mathbf{Z}^{\mathrm{mp}(j)})}_{(*)} \, \mathrm{d}\mathbf{Z}.$$

Proposed Method for ADMG

- Recall topological ADMG factorization: $p(\mathbf{Z}) = \prod_{j=1}^{D} p_{j|\mathrm{mp}(j)}(Z^j|\mathbf{Z}^{\mathrm{mp}(j)})$.

- Approximate each conditional by kernel-based estimator. Let $K^j : \mathcal{Z}^{\mathrm{mp}(j)} \to \mathbb{R}_{\geq 0}$ and

$$p(\mathbf{Z}) \simeq \prod_{j=1}^{D} \hat{p}_{j|\mathrm{mp}(j)}(Z^j|\mathbf{Z}^{\mathrm{mp}(j)}) \boxed{:= \frac{\sum_{i=1}^{n} \delta_{Z_i^j}(Z^j) K^j(\mathbf{Z}^{\mathrm{mp}(j)} - \mathbf{Z}_i^{\mathrm{mp}(j)})}{\sum_{k=1}^{n} K^j(\mathbf{Z}^{\mathrm{mp}(j)} - \mathbf{Z}_k^{\mathrm{mp}(j)})}}$$

Empirical conditional density

- Plug-in risk estimator

$$\hat{R}_{\mathrm{aug}}(f) = \int_{\mathcal{Z}} \ell(f, \mathbf{Z}) \prod_{j=1}^{D} \hat{p}_{j|\mathrm{mp}(j)}(Z^j|\mathbf{Z}^{\mathrm{mp}(j)}) \mathrm{d}\mathbf{Z} \boxed{= \sum_{\boldsymbol{i} \in [n]^D} \hat{w}_{\boldsymbol{i}} \cdot \ell(f, \mathbf{Z}_{\boldsymbol{i}})}$$

Augmented data + instance weights

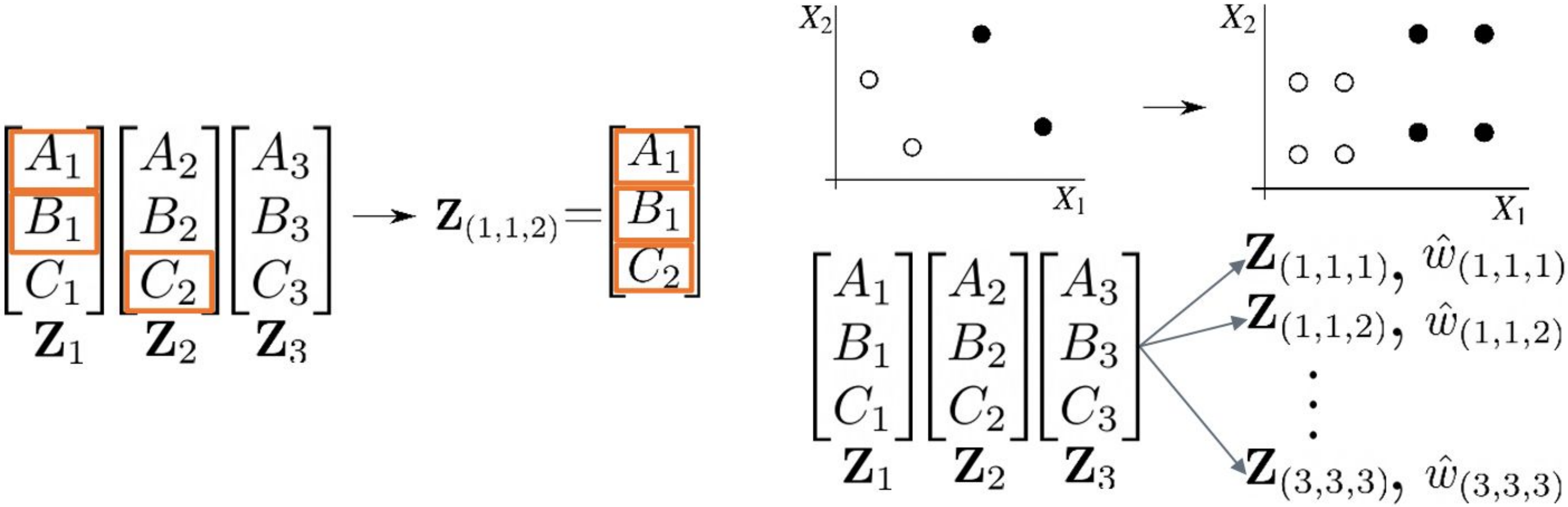
- Considering all the possible resampling candidates
- instance-weighted data augmentation procedure:

$$\hat{R}_{\mathrm{aug}}(f) = \sum_{\boldsymbol{i} \in [n]^D} \hat{w}_{\boldsymbol{i}} \cdot \ell(f, \mathbf{Z}_{\boldsymbol{i}}),$$

$$\hat{w}_{\boldsymbol{i}} = \prod_{j=1}^{D} \frac{K^j(\mathbf{Z}_{\boldsymbol{i}_{1:j-1}}^{\mathrm{mp}(j)} - \mathbf{Z}_i^{\mathrm{mp}(j)})}{\sum_{k=1}^{n} K^j(\mathbf{Z}_{\boldsymbol{i}_{1:j-1}}^{\mathrm{mp}(j)} - \mathbf{Z}_k^{\mathrm{mp}(j)})},$$

$$\mathbf{Z}_{\boldsymbol{i}} = (Z_{i_1}^1, \ldots, Z_{i_D}^D), \quad \mathbf{Z}_{\boldsymbol{i}_{1:j-1}} = (Z_{i_1}^1, \ldots, Z_{i_{j-1}}^{j-1}),$$

$$\hat{f} \in \arg\min_{f \in \mathcal{F}} \{(1-\lambda)\hat{R}_{\mathrm{emp}}(f) + \lambda \hat{R}_{\mathrm{aug}}(f) + \Omega(f)\}$$



A_1 A_2 A_3
B_1 B_2 B_3
C_1 C_2 C_3
$\mathbf{Z}_1$ $\mathbf{Z}_2$ $\mathbf{Z}_3$
$\mathbf{Z}_{(1,1,2)} = [A_1, B_1, C_2]$
X_2
X_1
$\mathbf{Z}_{(1,1,1)}, \hat{w}_{(1,1,1)}$
$\mathbf{Z}_{(1,1,2)}, \hat{w}_{(1,1,2)}$
$\mathbf{Z}_{(3,3,3)}, \hat{w}_{(3,3,3)}$

In Practice: Compute the weights

- Construct the probability tree

$$\hat{w}_{\boldsymbol{i}_{1:0}} = 1, \quad \hat{w}_{\boldsymbol{i}_{1:j}} = \hat{w}_{i_j|\boldsymbol{i}_{1:j-1}} \cdot \hat{w}_{\boldsymbol{i}_{1:j-1}} \; (j \in [D], \boldsymbol{i}_{1:j-1} \in [n]^{j-1}),$$

$$\hat{w}_{i_j|\boldsymbol{i}_{1:j-1}} := \frac{K^j(\mathbf{Z}^{\mathfrak{mp}(j)}_{\boldsymbol{i}_{1:j-1}} - \mathbf{Z}^{\mathfrak{mp}(j)}_{i})}{\sum_{i=1}^{n} K^j(\mathbf{Z}^{\mathfrak{mp}(j)}_{\boldsymbol{i}_{1:j-1}} - \mathbf{Z}^{\mathfrak{mp}(j)}_{i})},$$

Kernel Function:

$$K^j(\boldsymbol{x}-\boldsymbol{y}) := \prod_{j' \in \mathrm{mp}(j)} \frac{1}{\boldsymbol{h}^{j'}} K^j_{j'}\left(\frac{\boldsymbol{x}^{j'}-\boldsymbol{y}^{j'}}{\boldsymbol{h}^{j'}}\right)$$

Continuous: $K^j_{j'}(x-y) := (2\pi)^{-1/2} \exp\left(-\frac{(x-y)^2}{2}\right).$

Discrete: $K^j_{j'}(x-y) := \mathbb{1}[x=y]$

Algorithm

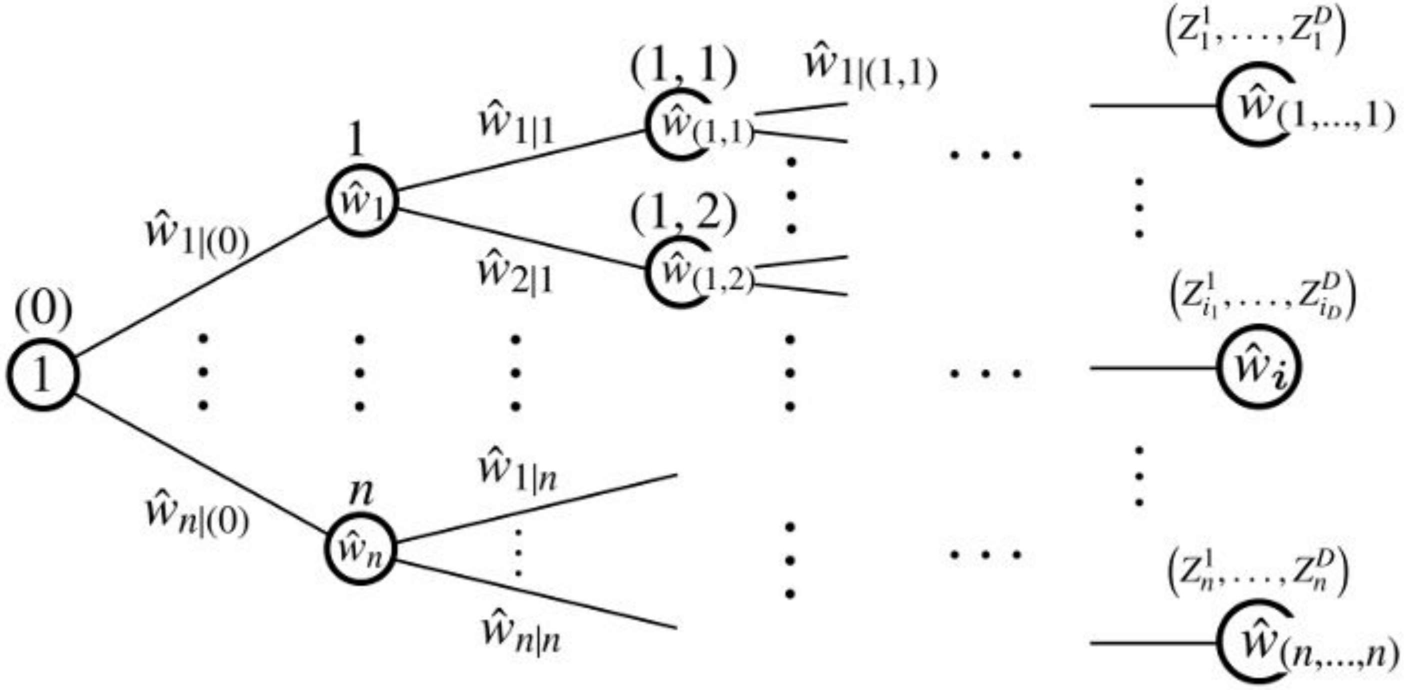
Algorithm 1 Proposed method: ADMG data augmentation

Input: Training data $\mathcal{D}$, ADMG $\hat{\mathcal{G}}$, coefficient $\lambda \in [0, 1]$, regularization functional Ω, pruning threshold $\theta \in [0, 1)$, hypothesis class $\mathcal{F}$, kernel functions $\{K^j\}_{j=1}^D$, loss function ℓ.

```
function FillProbTree(𝒟, Ĝ, θ, {K^j}_{j=1}^D)        ▷ see Fig. 2
    for j ∈ [D]                                        ▷ for each variable j
        for i_{1:j−1} ∈ [n]^{j−1}                      ▷ current node (depth j)
            for i_j ∈ [n]                              ▷ next node (depth j + 1)
                ŵ_{i_{1:j−1}} ← ŵ_{i_{1:j−1}} 𝟙[ŵ_{i_{1:j−1}} ≥ θ]     ▷ pruning
                ŵ_{i_{1:j}} ← ŵ_{i_j | i_{1:j−1}} · ŵ_{i_{1:j−1}}
    return 𝒲_aug := {ŵ_i}_{i∈[n]^D}
Let 𝒲_aug = FillProbTree(𝒟, Ĝ, θ, {K^j}_{j=1}^D).
Let R̂_aug(f) := Σ_{i∈[n]^D} ŵ_i · ℓ(f, Z_i).
Let R̃_λ(f) := (1 − λ)R̂_emp(f) + λR̂_aug(f) + Ω(f).
```

Output: $\hat{f} \in \underset{f \in \mathcal{F}}{\arg\min}\, \tilde{R}_\lambda(f)$: the predictor.

Algorithm



$\left(Z_1^1, \ldots, Z_1^D\right)$
$(1,1)$
$\hat{w}_{1|(1,1)}$
$\hat{w}_{1|1}$
$\hat{w}_{(1,1)}$
$\hat{w}_{(1,\ldots,1)}$
1
$\hat{w}_1$
$(1,2)$
$\hat{w}_{1|(0)}$
$\hat{w}_{2|1}$
$\hat{w}_{(1,2)}$
$\left(Z_{i_1}^1, \ldots, Z_{i_D}^D\right)$
(0)
1
$\hat{w}_i$
$\hat{w}_{1|n}$
n
$\hat{w}_{n|(0)}$
$\hat{w}_n$
$\left(Z_n^1, \ldots, Z_n^D\right)$
$\hat{w}_{n|n}$
$\hat{w}_{(n,\ldots,n)}$

Real World Data Experiment

- Goal: Confirm that the proposed method contributes to the performance of the trained predictor, especially in the small-data regime.
- **Compared:**

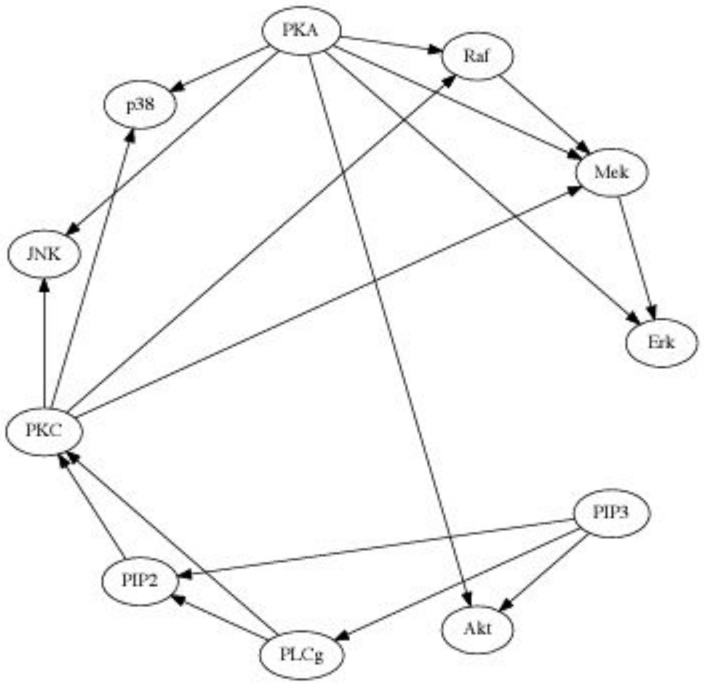
$$\hat{f} \in \underset{f \in \mathcal{F}}{\arg\min}\{\hat{R}_{\mathrm{emp}}(f) + \Omega(f)\}$$

$$\hat{f} \in \underset{f \in \mathcal{F}}{\arg\min}\{(1-\lambda)\hat{R}_{\mathrm{emp}}(f) + \lambda\hat{R}_{\mathrm{aug}}(f) + \Omega(f)\}$$

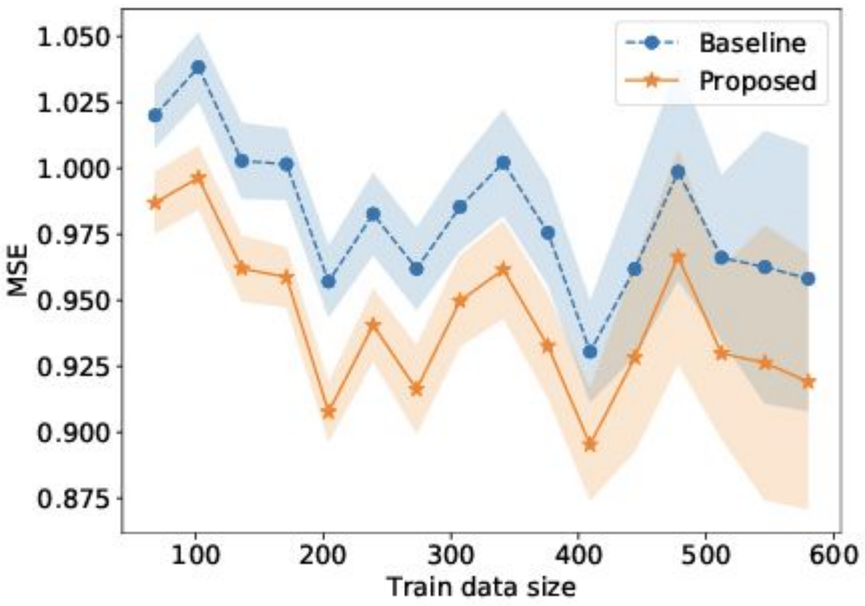
NAME	#VAR	#OBS	GRAPH
Sachs	11	853	Consensus
GSS	6	1380	Domain
Boston Housing	14	506	LiNGAM
Auto MPG	7	392	LiNGAM
White Wine	12	4898	LiNGAM
Red Wine	12	1599	LiNGAM

Real World Data Experiment

Sachs : Continuous, flow cytometry of proteins and phospholipids in human immune system cells



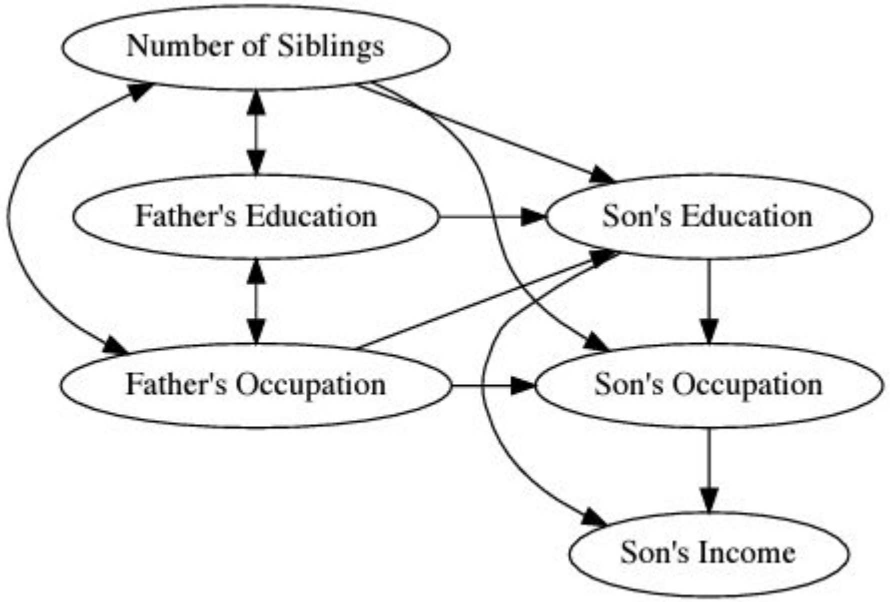
(a) Reference graph for Sachs data.



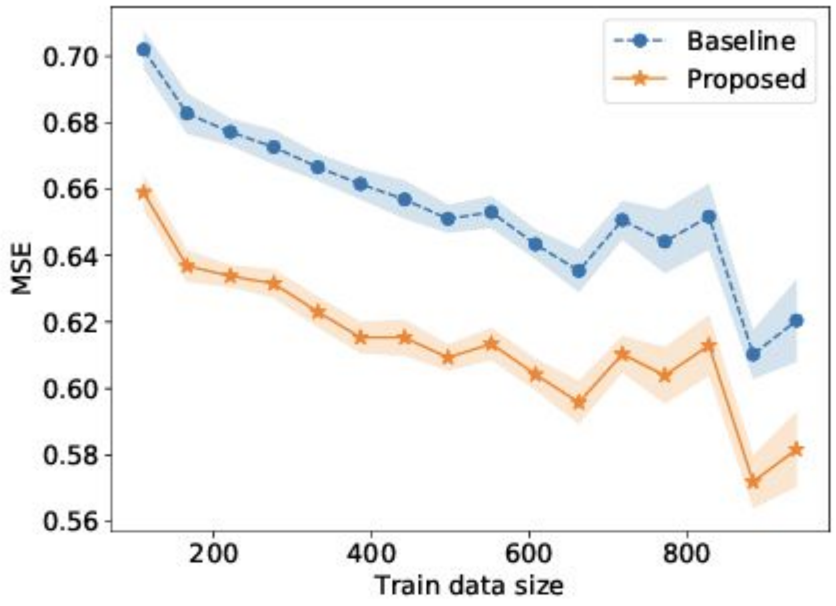
(a) Sachs data.

Real World Data Experiment

GSS: Part of General Social Survey

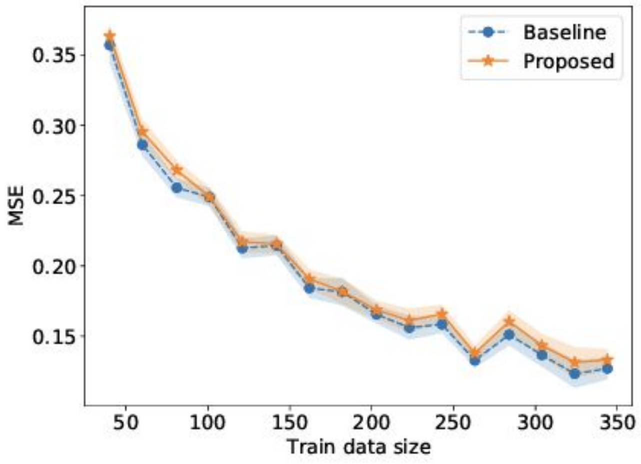


(b) Reference graph for GSS data.

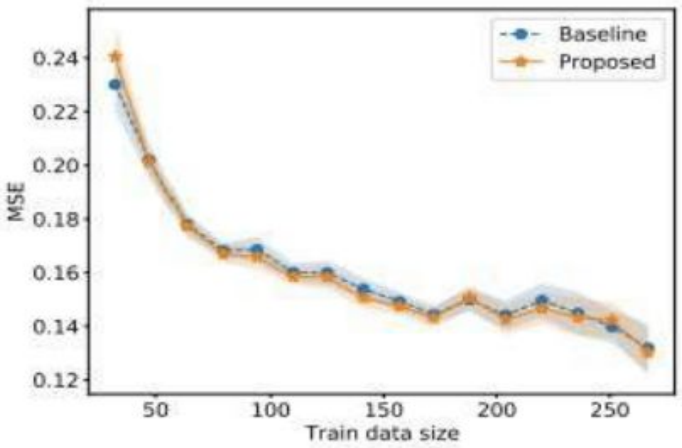


(b) GSS data.

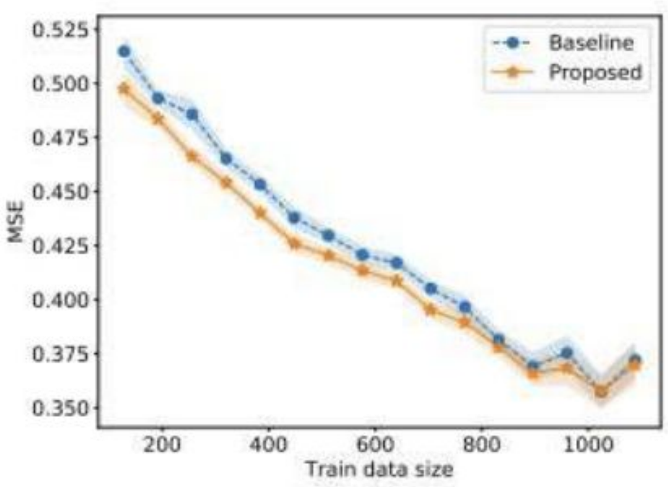
Real World Data Experiment



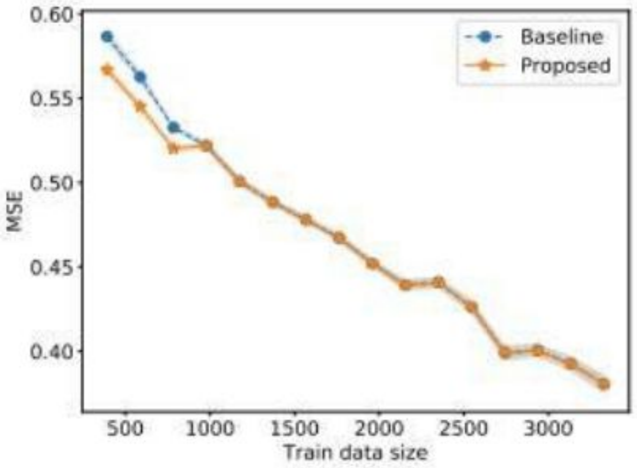
(c) Boston Housing data.



(d) Auto MPG data.



(e) Red Wine data.



(f) White Wine data.

Conclusion

- Proposed a data augmentation method to use the causal graphical prior knowledge in predictive modeling.
- Experimentally, the benefit may be worth the extra complexity and bias in small-data regime when domain knowledge is available.

Question

- What is the difference between **markov pillow** and **markov blanket**?

Theorem (Excess Risk Bound; informal)

$\hat{f} \in \arg\min_{f \in \mathcal{F}}\{\hat{R}_{\mathrm{aug}}(f)\},\ f^* \in \arg\min_{f \in \mathcal{F}}\{R(f)\}$

$$\boxed{R(\hat{f}) - R(f^*) \le \underbrace{C_1 R_{\mathbf{H}} + C_p}_{\text{Kernel Bias}} + \underbrace{C_2 R_K + C_3 R_{\mathcal{F},K}}_{\text{Complexity terms}} + \underbrace{C_4 \sqrt{\frac{\log(4D/\delta)}{2n}}}_{\text{Uncertainty}}}$$

w/ high probability.

- The complexity terms have a better sample-size dependency than the usual Rademacher complexity, implying mitigated overfitting. (Intuition: Synthesized data ⇨ Reduced possibility of overfitting.)
- But the bias due to the kernel approximation is introduced.