Online Graph Learning In Dynamic Environments

Xiang Zhang[1] Qiao Wang[1]

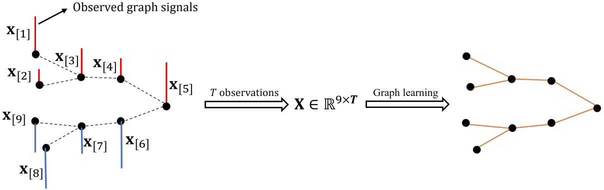
[1]School of Information Science and Engineering
Southeast University

August 30, 2022

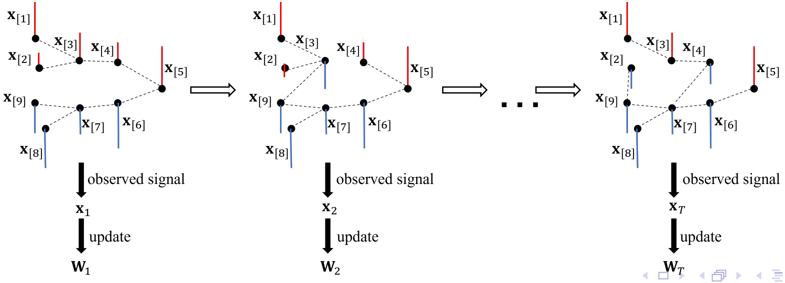
Contents

Contents

- Graph learning task



- Online graph learning scenario



- Batch graph learning based on smoothness assumption [Kal16]

$$\min_{\mathbf{L}\in\mathcal{L}} \operatorname{Tr}\left(\mathbf{X}^\top \mathbf{L}\mathbf{X}\right) - \alpha \mathbf{1}^\top \log\left(\operatorname{diag}(\mathbf{L})\right) + \frac{\beta}{2}\|\operatorname{diag}_0(\mathbf{L})\|_{\mathrm{F}}^2,$$
$$= \min_{\mathbf{w}\geq 0} 2\mathbf{z}^\top \mathbf{w} - \alpha \mathbf{1}^\top \log(\mathbf{S}\mathbf{w}) + \beta\|\mathbf{w}\|_2^2 = \min_{\mathbf{w}\geq 0} f(\mathbf{w})$$

- Online version of smoothness based graph learning [SMC21]
  - Update pairwise distance vector (data)

$$\bar{\mathbf{z}}_t = \gamma \bar{\mathbf{z}}_{t-1} + (1-\gamma)\mathbf{z}_t$$

  - Update new graph

$$\mathbf{w}_{t+1} = \prod_{\mathcal{W}_v}\left(\mathbf{w}_t - \eta_t \nabla f_t(\mathbf{w}_t)\right),$$

where $f_t(\mathbf{w}; \mathbf{x}_1, ..., \mathbf{x}_t) \triangleq 2\bar{\mathbf{z}}_t^\top \mathbf{w} - \alpha \mathbf{1}^\top \log(\mathbf{S}\mathbf{w}) + \beta\|\mathbf{w}\|_2^2$

- Limitation

Totally ignore the evolution of dynamic graphs !

- Batch graph learning based on smoothness assumption [Kal16]

$$\min_{\mathbf{L}\in\mathcal{L}} \operatorname{Tr}\left(\mathbf{X}^\top \mathbf{L}\mathbf{X}\right) - \alpha \mathbf{1}^\top \log\left(\operatorname{diag}(\mathbf{L})\right) + \frac{\beta}{2}\|\operatorname{diag}_0(\mathbf{L})\|_F^2,$$
$$= \min_{\mathbf{w}\geq 0} 2\mathbf{z}^\top \mathbf{w} - \alpha \mathbf{1}^\top \log(\mathbf{S}\mathbf{w}) + \beta\|\mathbf{w}\|_2^2 = \min_{\mathbf{w}\geq 0} f(\mathbf{w})$$

- Online version of smoothness based graph learning [SMC21]
  - Update pairwise distance vector (data)

$$\bar{\mathbf{z}}_t = \gamma \bar{\mathbf{z}}_{t-1} + (1-\gamma)\mathbf{z}_t$$

  - Update new graph

$$\mathbf{w}_{t+1} = \prod_{\mathcal{W}_v}\left(\mathbf{w}_t - \eta_t \nabla f_t(\mathbf{w}_t)\right),$$

where $f_t(\mathbf{w}; \mathbf{x}_1, ..., \mathbf{x}_t) \triangleq 2\bar{\mathbf{z}}_t^\top \mathbf{w} - \alpha \mathbf{1}^\top \log(\mathbf{S}\mathbf{w}) + \beta\|\mathbf{w}\|_2^2$

- Limitation

Totally ignore the evolution of dynamic graphs !

- Batch graph learning based on smoothness assumption [Kal16]

$$\min_{\mathbf{L}\in\mathcal{L}} \operatorname{Tr}\left(\mathbf{X}^\top\mathbf{L}\mathbf{X}\right) - \alpha\mathbf{1}^\top \log\left(\operatorname{diag}(\mathbf{L})\right) + \frac{\beta}{2}\|\operatorname{diag}_0(\mathbf{L})\|_{\mathrm{F}}^2,$$
$$=\min_{\mathbf{w}\geq 0} 2\mathbf{z}^\top\mathbf{w} - \alpha\mathbf{1}^\top\log(\mathbf{S}\mathbf{w}) + \beta\|\mathbf{w}\|_2^2 = \min_{\mathbf{w}\geq 0} f(\mathbf{w})$$

- Online version of smoothness based graph learning [SMC21]
  - Update pairwise distance vector (data)

$$\bar{\mathbf{z}}_t = \gamma\bar{\mathbf{z}}_{t-1} + (1-\gamma)\mathbf{z}_t$$

  - Update new graph

$$\mathbf{w}_{t+1} = \prod_{\mathcal{W}_v}\left(\mathbf{w}_t - \eta_t\nabla f_t(\mathbf{w}_t)\right),$$

where $f_t(\mathbf{w};\mathbf{x}_1,\ldots,\mathbf{x}_t) \triangleq 2\bar{\mathbf{z}}_t^\top\mathbf{w} - \alpha\mathbf{1}^\top\log(\mathbf{S}\mathbf{w}) + \beta\|\mathbf{w}\|_2^2$

- Limitation

Totally ignore the evolution of dynamic graphs !

Contents

Proposed Algorithm

- Basic assumption
  Graphs are assumed to evolve with a dynamic model Φ, i.e.,

$$\mathbf{w}_{t+1} = \Phi(\mathbf{w}_t)$$

- Some examples
  - Social network [Sni01]
    If $|\mathbf{W}_{ik^*}\mathbf{W}_{jk^*}| > |\mathbf{W}_{ij}|$

$$(\Phi(\mathbf{W}))_{[ij]} = (1-a)\mathbf{W}_{[ij]} + a\mathbf{W}_{[ik^*]}\mathbf{W}_{[jk^*]},$$

    where $k^* = \mathrm{argmax}_k\, |\mathbf{W}_{[ik^*]}\mathbf{W}_{[jk^*]}|$
  - Autoregressive (AR) model [HW15]

$$\Phi(\mathbf{w}_t) = \mathbf{A}\mathbf{w}_t$$

  - Transition mode

$$\Phi(\mathbf{w}_t) = a\mathbf{w}_t + (1-a)\mathbf{w}_{\mathrm{target}}$$

Proposed Algorithm

- Basic assumption
  Graphs are assumed to evolve with a dynamic model Φ, i.e.,

$$\mathbf{w}_{t+1} = \Phi(\mathbf{w}_t)$$

- Some examples
  - Social network [Sni01]
    If $|\mathbf{W}_{ik^*}\mathbf{W}_{jk^*}| > |\mathbf{W}_{ij}|$

$$(\Phi(\mathbf{W}))_{[ij]} = (1-a)\mathbf{W}_{[ij]} + a\mathbf{W}_{[ik^*]}\mathbf{W}_{[jk^*]},$$

    where $k^* = \mathrm{argmax}_k \, |\mathbf{W}_{[ik^*]}\mathbf{W}_{[jk^*]}|$
  - Autoregressive (AR) model [HW15]

$$\Phi(\mathbf{w}_t) = \mathbf{A}\mathbf{w}_t$$

  - Transition mode

$$\Phi(\mathbf{w}_t) = a\mathbf{w}_t + (1-a)\mathbf{w}_{\text{target}}$$

Proposed Algorithm

- No prior knowledge about Φ
  A data-driven method proposed by [SDP+20]

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}\in\mathcal{W}_v} \Big\{ \frac{1}{2}\mathbf{w}^\top \nabla_{\mathbf{ww}} f_t(\mathbf{w}_t)\mathbf{w} + \big(\nabla_{\mathbf{w}} f_t(\mathbf{w}_t) + h\nabla_{t\mathbf{w}} f_t(\mathbf{w}_t) - \nabla_{\mathbf{ww}} f_t(\mathbf{w}_t)\mathbf{w}_t\big)^\top \mathbf{w} \Big\}$$

- Full update procedure
  - Update pairwise distance vector (data)

$$\bar{\mathbf{z}}_t = \gamma\bar{\mathbf{z}}_{t-1} + (1-\gamma)\mathbf{z}_t$$

  - Update new graph

$$\breve{\mathbf{w}}_t = \prod_{\mathcal{W}_v}(\mathbf{w}_t - \eta_t \nabla f_t(\mathbf{w}_t)),$$

  - One more prediction

$$\mathbf{w}_{t+1} = \Phi(\breve{\mathbf{w}}_t)$$

Proposed Algorithm

- No prior knowledge about Φ
  A data-driven method proposed by [SDP+20]

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}\in\mathcal{W}_v}\Big\{\frac{1}{2}\mathbf{w}^\top\nabla_{\mathbf{ww}} f_t(\mathbf{w}_t)\mathbf{w} + \big(\nabla_{\mathbf{w}} f_t(\mathbf{w}_t) + h\nabla_{t\mathbf{w}} f_t(\mathbf{w}_t) - \nabla_{\mathbf{ww}} f_t(\mathbf{w}_t)\mathbf{w}_t\big)^\top\mathbf{w}\Big\}$$

- Full update procedure
  - Update pairwise distance vector (data)

$$\bar{\mathbf{z}}_t = \gamma\bar{\mathbf{z}}_{t-1} + (1-\gamma)\mathbf{z}_t$$

  - Update new graph

$$\breve{\mathbf{w}}_t = \prod_{\mathcal{W}_v}(\mathbf{w}_t - \eta_t\nabla f_t(\mathbf{w}_t)),$$

  - One more prediction

$$\mathbf{w}_{t+1} = \Phi(\breve{\mathbf{w}}_t)$$

Contents

Theoretic analysis

- Definition of dynamic regret

$$Reg_d(T) \triangleq \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{w}_t^*)$$

- Main result

Theorem

If the selected stepsize η_t is a constant satisfying $\eta_t = \eta \leq \left(2\beta + \alpha(d-1)/deg_{\min}^2\right)^{-1}$, the dynamic regret Reg_d of the proposed algorithm satisfies

$$Reg_d(T) \leq \frac{d(d-1)w_{\max}^2}{4\eta} + \frac{\sqrt{2d(d-1)}w_{\max}}{2\eta} C_V^d + \frac{\eta T}{2} L^2$$

where $C_V^d \triangleq \sum_{t=2}^{T} \|\mathbf{w}_t^* - \Phi(\mathbf{w}_{t-1}^*)\|_2$, and $w_{\max}, L$, are some predefined constants

- Discussions
  - If $\mathcal{O}(1/\sqrt{T})$, sublinear regret is obtained
  - If Φ is selected appropriately, $C_V^d \leq C_V \triangleq \sum_{t=2}^{T} \|\mathbf{w}_t^* - \mathbf{w}_{t-1}^*\|_2$, where C_V measures the variations of dynamic environments.

- Definition of dynamic regret

$$Reg_d(T) \triangleq \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{w}_t^*)$$

- Main result

Theorem

If the selected stepsize η_t is a constant satisfying $\eta_t = \eta \leq \left(2\beta + \alpha(d-1)/deg_{\min}^2\right)^{-1}$, the dynamic regret Reg_d of the proposed algorithm satisfies

$$Reg_d(T) \leq \frac{d(d-1)w_{\max}^2}{4\eta} + \frac{\sqrt{2d(d-1)}w_{\max}}{2\eta} C_V^d + \frac{\eta T}{2} L^2$$

where $C_V^d \triangleq \sum_{t=2}^{T} \|\mathbf{w}_t^* - \Phi(\mathbf{w}_{t-1}^*)\|_2$, and $w_{\max}, L$, are some predefined constants

- Discussions
  - If $\mathcal{O}(1/\sqrt{T})$, sublinear regret is obtained
  - If Φ is selected appropriately, $C_V^d \leq C_V \triangleq \sum_{t=2}^{T} \|\mathbf{w}_t^* - \mathbf{w}_{t-1}^*\|_2$, where C_V measures the variations of dynamic environments.

Theoretic analysis

- Definition of dynamic regret

$$Reg_d(T) \triangleq \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{w}_t^*)$$

- Main result

Theorem

If the selected stepsize η_t is a constant satisfying $\eta_t = \eta \leq \left(2\beta + \alpha(d-1)/deg_{\min}^2\right)^{-1}$, the dynamic regret Reg_d of the proposed algorithm satisfies

$$Reg_d(T) \leq \frac{d(d-1)w_{\max}^2}{4\eta} + \frac{\sqrt{2d(d-1)}w_{\max}}{2\eta} C_V^d + \frac{\eta T}{2} L^2$$

where $C_V^d \triangleq \sum_{t=2}^{T} \|\mathbf{w}_t^* - \Phi(\mathbf{w}_{t-1}^*)\|_2$, and $w_{\max}, L,$ are some predefined constants

- Discussions
  - If $\mathcal{O}(1/\sqrt{T})$, sublinear regret is obtained
  - If Φ is selected appropriately, $C_V^d \leq C_V \triangleq \sum_{t=2}^{T} \|\mathbf{w}_t^* - \mathbf{w}_{t-1}^*\|_2$, where C_V measures the variations of dynamic environments.

Contents

Experiments

- Main settings
  - Graph generation

    An initial graph $\mathbf{w}_0$ is generated by following [DTFV16], and the remaining graphs are generated by $\mathbf{w}_{t+1} = \Phi(\mathbf{w}_t)$
  - Graph signal generation [DTFV16]

$$\mathbf{x}_t \sim Gauss\left(0, \left(\mathbf{L}_t + \sigma_w^2 \mathbf{I}\right)^{\dagger}\right)$$

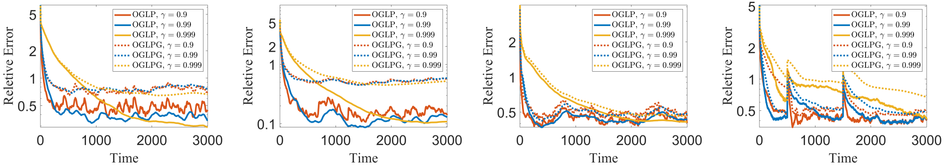
  - Evaluation metrics

$$\text{Relative error} = \frac{\|\mathbf{W}_t - \mathbf{W}_t^*\|_{\mathrm{F}}}{\|\mathbf{W}_t^*\|_{\mathrm{F}}}$$

  - Baseline

    The algorithm without prior evolutionary patterns, named as OGLPG [SMC21]

- Synthetic data



(a) (b) (c) (d)

Figure: Relative error of different dynamic models (a) AR model (b) Transition model (c) Social network (d) Switching model

- Real data
  - Dataset

    (1) Data source: stock closing price data[1]

    (2) Task: Learning time-varying graphs of relationships among different companies

    (3) Data selection: data of 10 pharmaceutical companies from August 1st 2019 to July 30th 2021, during which COVID-19 pandemic outbroke
  - Main results

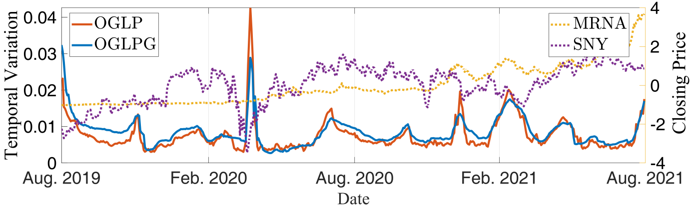


Figure: Temporal variation of stock market graph

[1] https://finance.yahoo.com/

Contents

Conclusions

- We propose an online graph learning method using dynamic evolutionary patterns
- A theoretic analysis is provided to prove the power of dynamic evolutionary patterns
- Experimental results of both synthetic and real data illustrate the superiority of our algorithm

The End

References I

[DTFV16] Xiaowen Dong, Dorina Thanou, Pascal Frossard, and Pierre Vandergheynst, *Learning laplacian matrix in smooth graph signal representations*, IEEE Transactions on Signal Processing **64** (2016), no. 23, 6160–6173.

[HW15] Eric C Hall and Rebecca M Willett, *Online convex optimization in dynamic environments*, IEEE Journal of Selected Topics in Signal Processing **9** (2015), no. 4, 647–662.

[Kal16] Vassilis Kalofolias, *How to learn a graph from smooth signals*, Artificial Intelligence and Statistics, PMLR, 2016, pp. 920–929.

[SDP$^+$20] Andrea Simonetto, Emiliano Dall'Anese, Santiago Paternain, Geert Leus, and Georgios B Giannakis, *Time-varying convex optimization: Time-structured algorithms and applications*, Proceedings of the IEEE **108** (2020), no. 11, 2032–2048.

[SMC21] Seyed Saman Saboksayr, Gonzalo Mateos, and Mujdat Cetin, *Online graph learning under smoothness priors*, 2021 29th European Signal Processing Conference (EUSIPCO), IEEE, 2021, pp. 1820–1824.

[Sni01] Tom AB Snijders, *The statistical evaluation of social network dynamics*, Sociological methodology **31** (2001), no. 1, 361–395.