Online Dictionary Learning from Dependent Data Samples and Networks

Hanbaek Lyu

Department of Mathematics, IFDS
University of Wisconsin - Madison

Partially supported by NSF DMS #2010035

Oct. 13 2021

Outline

Dictionary Learning

Introduction to Network Dictionary Learning

Stochastic Optimization and Online Matrix Factorization

Theory and Main results

Proof ideas

Future directions and some ongoing works

- Dictionary Learning: Learn r basis vectors from a given data set of 'vectors'

Learning parts of images (Olivetti face images)

- Dictionary Learning: Learn r basis vectors from a given data set of 'vectors'
  - 'vectors' may represent images, texts, time-serieses, graphs, etc.

- Dictionary Learning: Learn r basis vectors from a given data set of 'vectors'
  - 'vectors' may represent images, texts, time-serieses, graphs, etc.
  - Provides a compressed representation of complex objects using a few dictionary elements.

- Dictionary Learning: Learn r basis vectors from a given data set of 'vectors'
  - 'vectors' may represent images, texts, time-serieses, graphs, etc.
  - Provides a compressed representation of complex objects using a few dictionary elements.

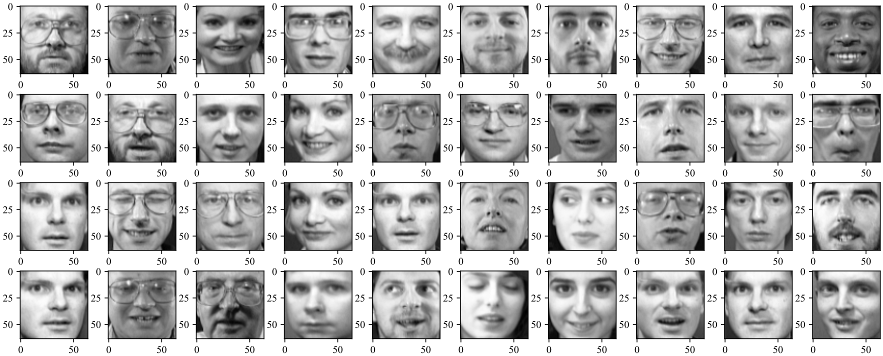


Figure: Sample images from the Olivetti face dataset (total 400 faces)

Learning parts of images (Olivetti face images)

- Dictionary Learning: Learn r basis vectors from a given data set of 'vectors'
  - 'vectors' may represent images, texts, time-serieses, graphs, etc.
  - Provides a compressed representation of complex objects using a few dictionary elements.

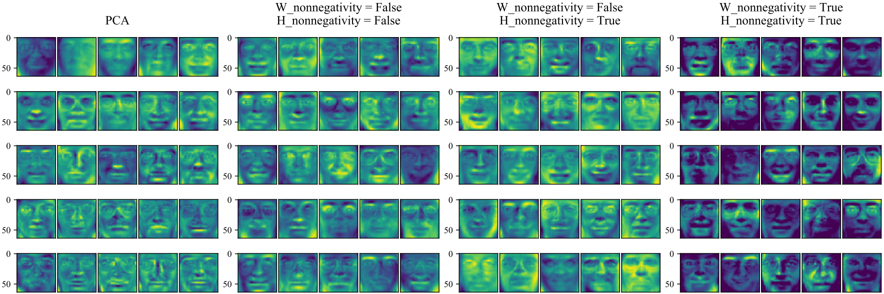


Figure: Example dictionaries learned by PCA and matrix factorization

Learning parts of images (MNIST handwritten digits)

- **Dictionary Learning**: Learn r basis vectors from a given data set of 'vectors'
  - 'vectors' may represent images, texts, time-serieses, graphs, etc.
  - Provides a compressed representation of complex objects using a few dictionary elements.

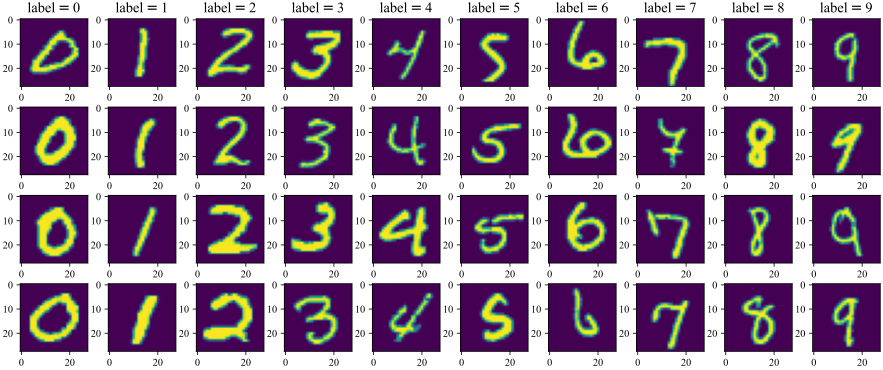


Figure: Sample MNIST images (total 70000 images of size 28× 28)

Learning parts of images (MNIST handwritten digits)

- Dictionary Learning: Learn r basis vectors from a given data set of 'vectors'
  - 'vectors' may represent images, texts, time-serieses, graphs, etc.
  - Provides a compressed representation of complex objects using a few dictionary elements.

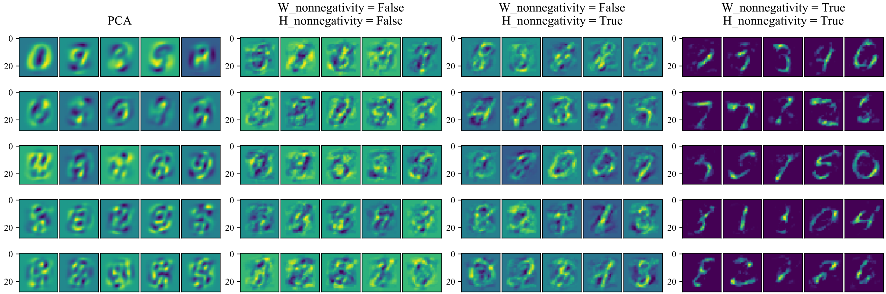


Figure: Example dictionaries learned by PCA and matrix factorization

Learning parts of images (20 News Grpups)

- Dictionary Learning: Learn r basis vectors from a given data set of 'vectors'
  - 'vectors' may represent images, texts, time-serieses, graphs, etc.
  - Provides a compressed representation of complex objects using a few dictionary elements.

```
>>>> data_cleaned[i] Anyone know what would cause my IIcx to not turn on when I hit the keyboard
switch?  The one in the back of the machine doesn't work either...
The only way I can turn it on is to unplug the machine for a few minutes,
then plug it back in and hit the power switch in the back immediately...
Sometimes this doesn't even work for a long time...

I remember hearing about this problem a long time ago, and that a logic
board failure was mentioned as the source of the problem...is this true?
```

Figure: Example of text data from the 20 News Groups (20 categories, 5616 articles)

- **Dictionary Learning**: Learn r basis vectors from a given data set of ‘vectors’
  - ‘vectors’ may represent images, texts, time-serieses, graphs, etc.
  - Provides a compressed representation of complex objects using a few dictionary elements.

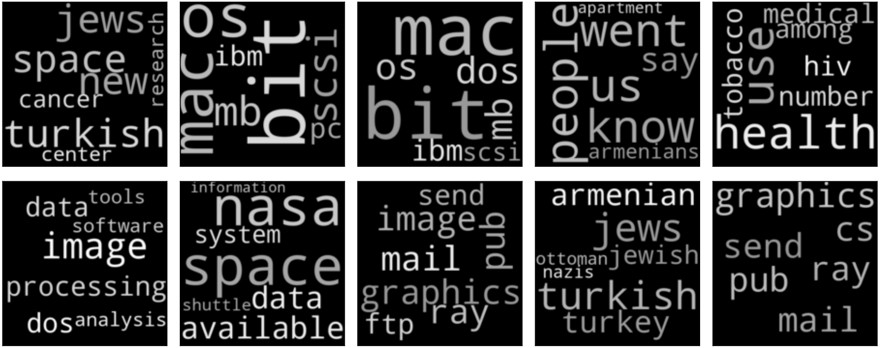


Figure: Example dictionaries (topics) learned by nonnegative matrix factorization from 20 News Groups

- **Dictionary Learning**: Learn r basis vectors from a given data set of 'vectors'
  - 'vectors' may represent images, texts, time-serieses, graphs, etc.
  - Provides a compressed representation of complex objects using a few dictionary elements.

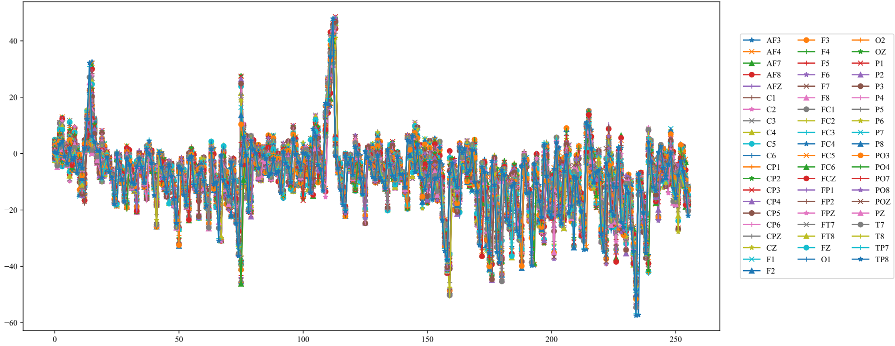


Figure: Brain EEG data from 61 electrodes (61-dimensional multivariate time-series)

- **Dictionary Learning**: Learn r basis vectors from a given data set of 'vectors'
  - 'vectors' may represent images, texts, time-serieses, graphs, etc.
  - Provides a compressed representation of complex objects using a few dictionary elements.

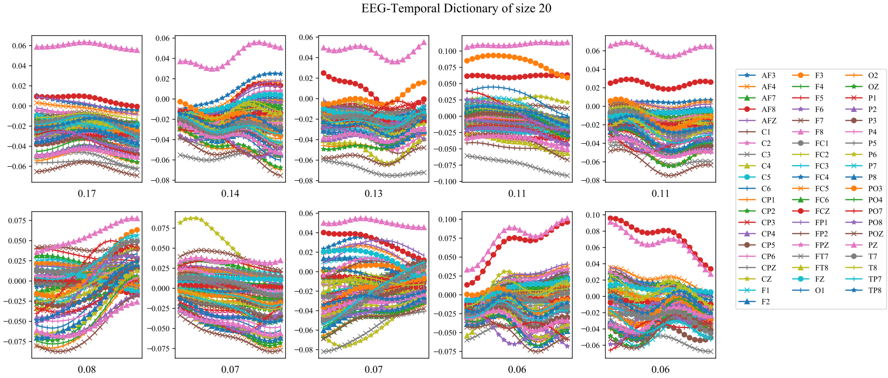
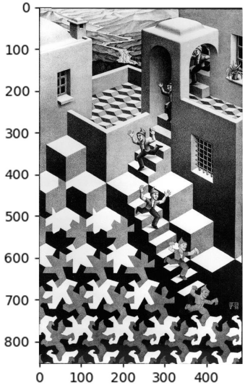


Figure: Temporal dictionary of window size $k = 20$ learned by matrix factorization

Learning parts of images – Image reconstruction

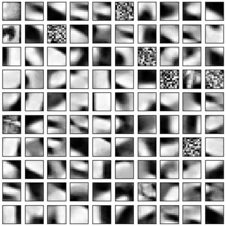
Original image -
Cycle by M. C. Escher (1928)



- Dictionary learning $\rightarrow$ Reconstruction, denoising, transfer learning, etc.

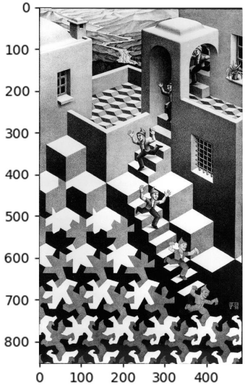
What do we do with dictionaries? – Image reconstruction

Dictionary learned from
Cycle by M. C. Escher



(basis)

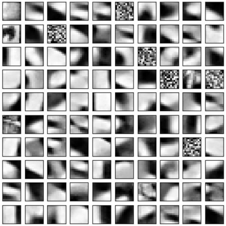
Original image -
Cycle by M. C. Escher (1928)



- Dictionary learning $\rightarrow$ Reconstruction, denoising, transfer learning, etc.

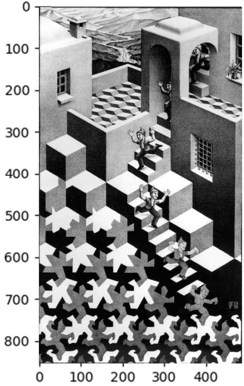
What do we do with dictionaries? – Image reconstruction

Dictionary learned from
Cycle by M. C. Escher

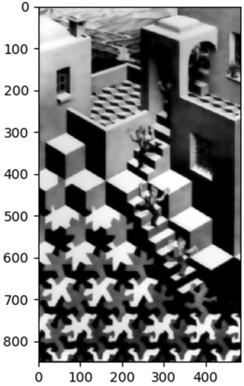


(basis)

Original image -
Cycle by M. C. Escher (1928)

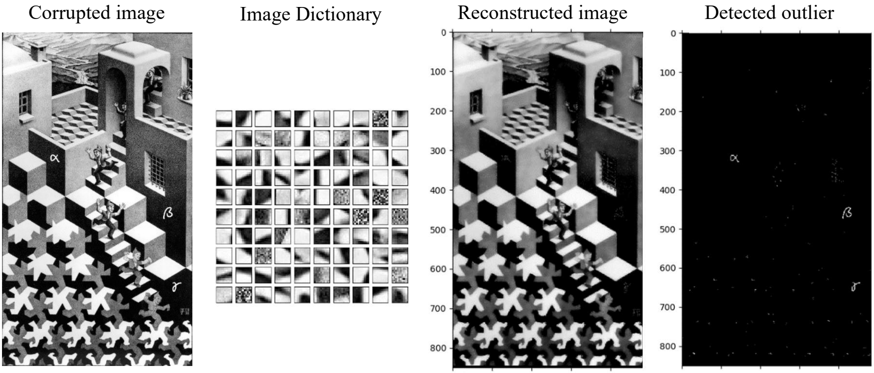


Reconstructed image
using learned dictionary



- Dictionary Learning $\rightarrow$ Reconstruction, denoising, transfer learning, etc.
- Img recons. = (local approx. by dict.) + (Averaging)

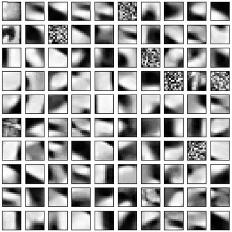
Learning parts of images – Image denoising



- Dictionary Learning → Reconstruction, denoising [1, 7], transfer learning, etc.
- Img recons. = (local approx. by dict.) + (Averaging)

Learning parts of images – Transfer learning

Dictionary learned from
Cycle by M. C. Escher



(basis)

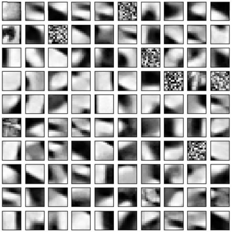
Original image -
Two Sisters by A. Renoir (1882)

(New data)

- Dictionary Learning → Reconstruction, denoising, transfer learning, etc.
- Img recons. = (local approx. by dict.) + (Averaging)

Learning parts of images – Transfer learning

Dictionary learned from
Cycle by M. C. Escher

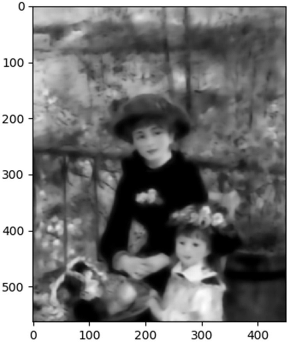


(basis)

Original image -
Two Sisters by A. Renoir (1882)

(New data)

Reconstructed image using Dict.
learned from Escher's Cycle

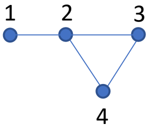


- Dictionary Learning $\rightarrow$ Reconstruction, denoising, transfer learning, etc.
- Img recons. = (local approx. by dict.) + (Averaging)

Outline

Networks: Basic language describing complex systems

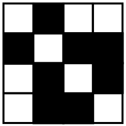
- In this talk: Simple networks (symmetric 0-1 matrices with 0's on diagonal)



Graph

	1	2	3	4
1	0	1	0	0
2	1	0	1	1
3	0	1	0	1
4	0	1	1	0

Matrix

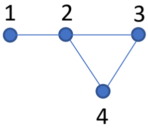


Pixel picture

Network Data

Networks: Basic language describing complex systems

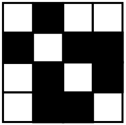
- In this talk: Simple networks (symmetric 0-1 matrices with 0's on diagonal)



Graph

$$
\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 1 \\ 3 & 0 & 1 & 0 & 1 \\ 4 & 0 & 1 & 1 & 0 \end{array}
$$

Matrix

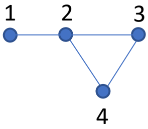


Pixel picture

- In pixel picture: Cross shape $\leftrightarrow$ hub node (node 2);
  Block shape $\leftrightarrow$ community (nodes 2,3,4)

Networks: Basic language describing complex systems

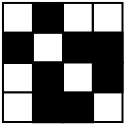
- In this talk: Simple networks (symmetric 0-1 matrices with 0's on diagonal)



Graph

	1	2	3	4
1	0	1	0	0
2	1	0	1	1
3	0	1	0	1
4	0	1	1	0

Matrix



Pixel picture

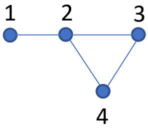
- In pixel picture: Cross shape $\leftrightarrow$ hub node (node 2); Block shape $\leftrightarrow$ community (nodes 2,3,4)

- Huge amount of information is being encoded into networks in various domains (e.g., Social networks, biological networks, brain networks, genetic networks, citation networks, ecology networks, economic networks, electric power networks, road networks)

Network Data

Networks: Basic language describing complex systems

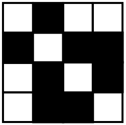
- In this talk: Simple networks (symmetric 0-1 matrices with 0's on diagonal)



Graph

$$
\begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}
$$

Matrix



Pixel picture

- In pixel picture: Cross shape $\leftrightarrow$ hub node (node 2);
  Block shape $\leftrightarrow$ community (nodes 2,3,4)

- Huge amount of information is being encoded into networks in various domains (e.g., Social networks, biological networks, brain networks, genetic networks, citation networks, ecology networks, economic networks, electric power networks, road networks)

- Developing proper theory and algorithm for network data analysis is becoming more important

Standard network summary

Stat	Coronavirus	SNAP FB	arXiv	Caltech	MIT	UCLA	Harvard
nodes	1555	4039	18772	769	6440	20467	15126
edges	4281	88234	198110	16656	251252	747613	824617
avg deg	3.19	43.69	21.10	43.31	78.02	73.05	109.033
edge density	0.002	0.01	0.001	0.05	0.01	0.003	0.007
avg clustering	0.039	0.60	0.63	0.40	0.27	0.21	0.21
diameter	9	8	14	6	8		

Standard network summary

Stat	Coronavirus	SNAP FB	arXiv	Caltech	MIT	UCLA	Harvard
nodes	1555	4039	18772	769	6440	20467	15126
edges	4281	88234	198110	16656	251252	747613	824617
avg deg	3.19	43.69	21.10	43.31	78.02	73.05	109.033
edge density	0.002	0.01	0.001	0.05	0.01	0.003	0.007
avg clustering	0.039	0.60	0.63	0.40	0.27	0.21	0.21
diameter	9	8	14	6	8		

- Standard network summary uses statistics based on either local or global properties of networks

Standard network summary

Stat	Coronavirus	SNAP FB	arXiv	Caltech	MIT	UCLA	Harvard
nodes	1555	4039	18772	769	6440	20467	15126
edges	4281	88234	198110	16656	251252	747613	824617
avg deg	3.19	43.69	21.10	43.31	78.02	73.05	109.033
edge density	0.002	0.01	0.001	0.05	0.01	0.003	0.007
avg clustering	0.039	0.60	0.63	0.40	0.27	0.21	0.21
diameter	9	8	14	6	8		

- Standard network summary uses statistics based on either local or global properties of networks
- Still not giving much information on **the structure of networks** at intermediate scales

Standard network summary

Stat	Coronavirus	SNAP FB	arXiv	Caltech	MIT	UCLA	Harvard
nodes	1555	4039	18772	769	6440	20467	15126
edges	4281	88234	198110	16656	251252	747613	824617
avg deg	3.19	43.69	21.10	43.31	78.02	73.05	109.033
edge density	0.002	0.01	0.001	0.05	0.01	0.003	0.007
avg clustering	0.039	0.60	0.63	0.40	0.27	0.21	0.21
diameter	9	8	14	6	8		

- Standard network summary uses statistics based on either local or global properties of networks
- Still not giving much information on **the structure of networks** at intermediate scales
- To overcome this problem, we develop a new way of summarizing networks based on analyzing k-node connected subgraphs

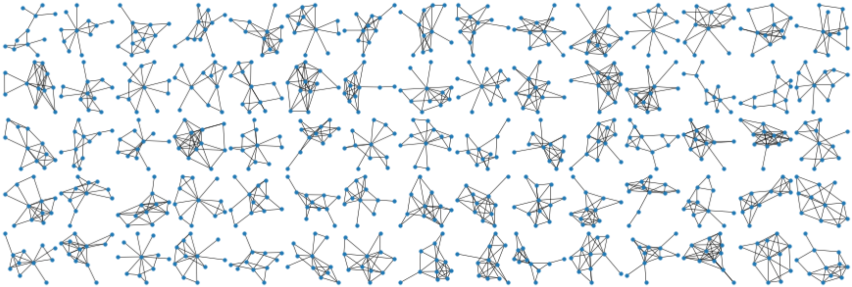
Dictionary learning with subgraphs?

- G = Caltech Facebook network (769 nodes, 16656 edges)

Dictionary learning with subgraphs?

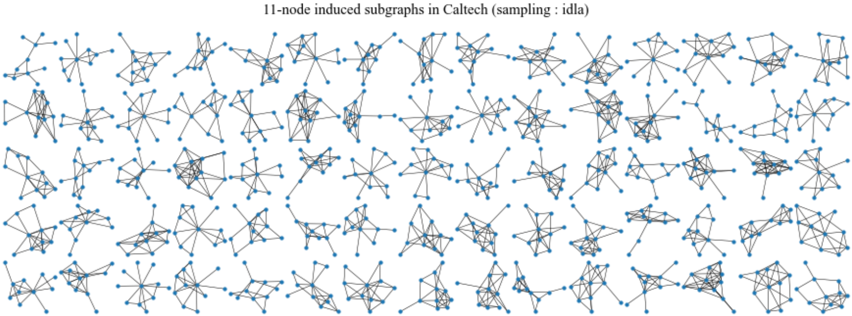
- $G =$ Caltech Facebook network (769 nodes, 16656 edges)
- Sample lots of k-node induced subgraphs from G

11-node induced subgraphs in Caltech (sampling : idla)



Dictionary learning with subgraphs?

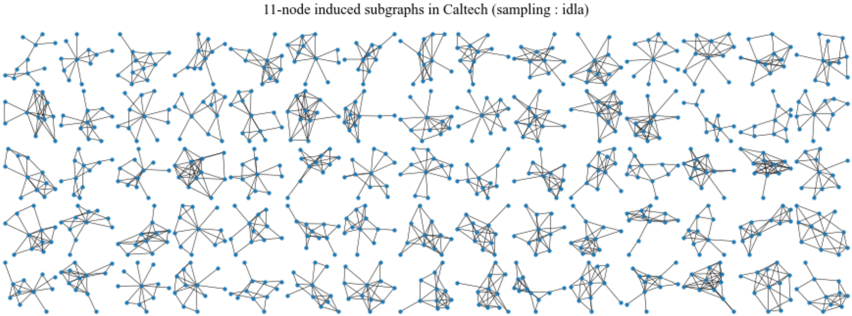
- $G =$ Caltech Facebook network (769 nodes, 16656 edges)
- Sample lots of k-node induced subgraphs from G



- There are LOTs of network sampling algorithms

Dictionary learning with subgraphs?

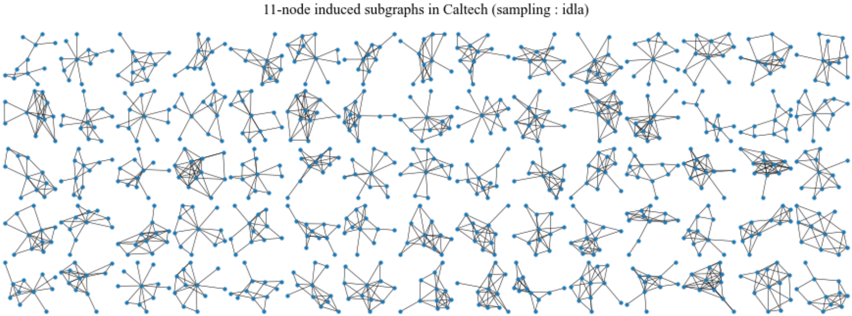
- $G =$ Caltech Facebook network (769 nodes, 16656 edges)
- Sample lots of k-node induced subgraphs from G



- There are LOTs of network sampling algorithms
- The above uses "sandpile sampling": Drop random walking particles on a chosen node until collecting k distinct nodes

Dictionary learning with subgraphs?

- $G =$ Caltech Facebook network (769 nodes, 16656 edges)
- Sample lots of k-node induced subgraphs from G

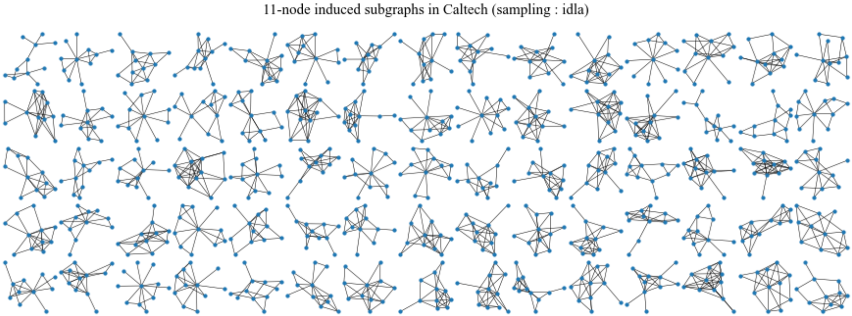


- There are LOTs of network sampling algorithms
- The above uses "sandpile sampling": Drop random walking particles on a chosen node until collecting k distinct nodes

- But how do we vectorize those subgraphs?

Dictionary learning with subgraphs?

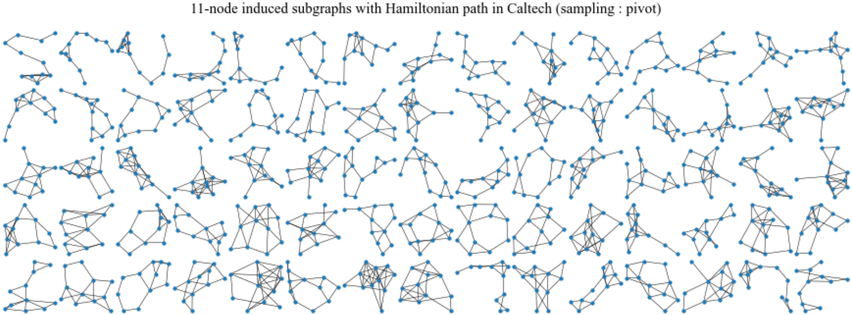
- $G =$ Caltech Facebook network (769 nodes, 16656 edges)
- Sample lots of k-node induced subgraphs from G



  - There are LOTs of network sampling algorithms
  - The above uses "sandpile sampling": Drop random walking particles on a chosen node until collecting k distinct nodes
- But how do we vectorize those subgraphs?
  - Adjacency matrix? Too many ways to order the nodes!

Dictionary learning with subgraphs?

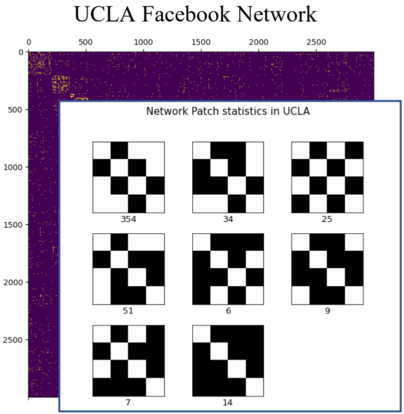
- $G =$ Caltech Facebook network (769 nodes, 16656 edges)
- Sample lots of k-node induced subgraphs from G with Hamiltonian paths

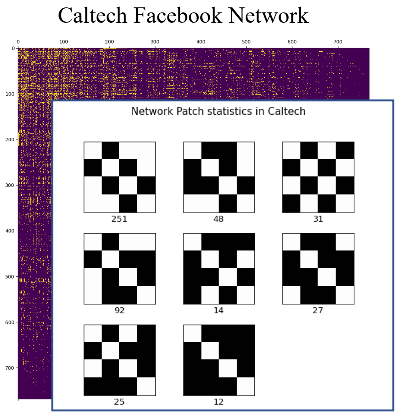


  - Choose a uniformly random k-path and take the induced subgraph
  - How? MCMC k-walk motif sampling + rejection sampling (will be discussed)

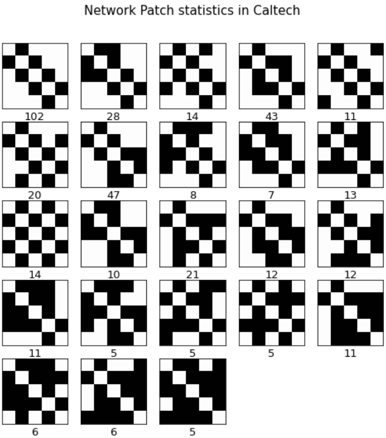
- But how do we vectorize those subgraphs?
  - Adjacency matrix w.r.t. the Hamiltonian path ordering

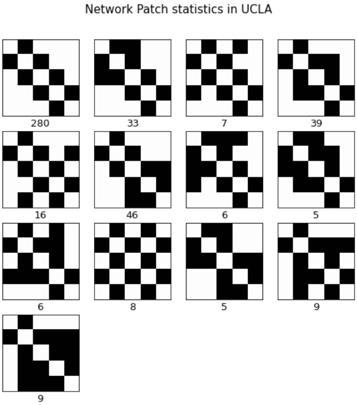
Statistics of k-node subgraph patterns



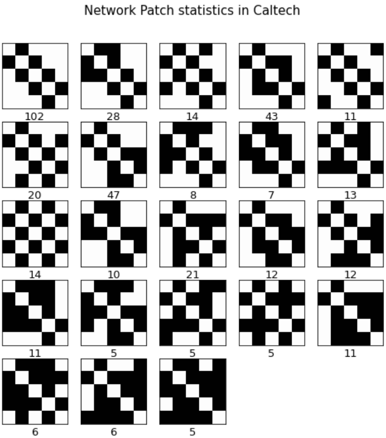


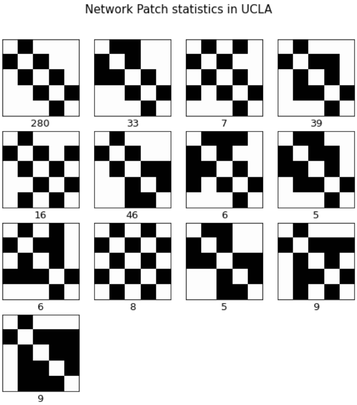
Statistics of k-node subgraph patterns





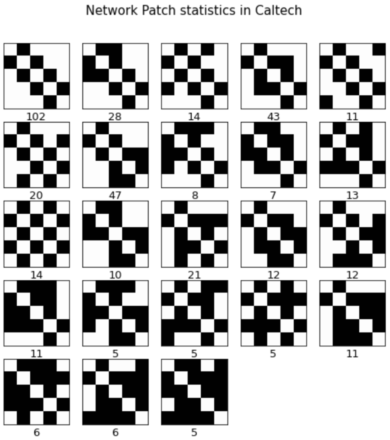
Statistics of k-node subgraph patterns

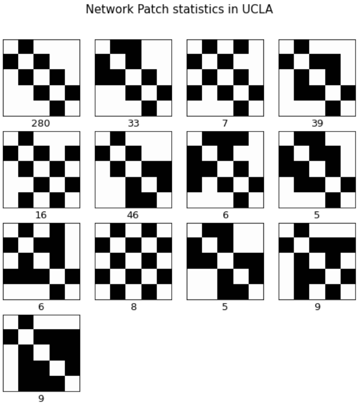




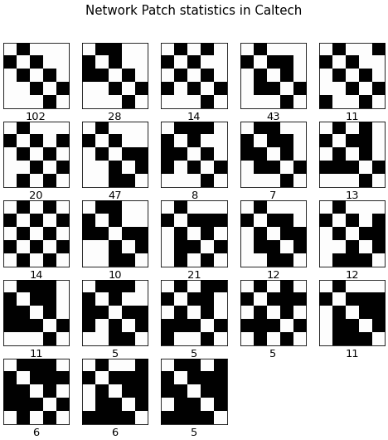
- What is the dimension of k-node connected subgraph patterns? (for large k)

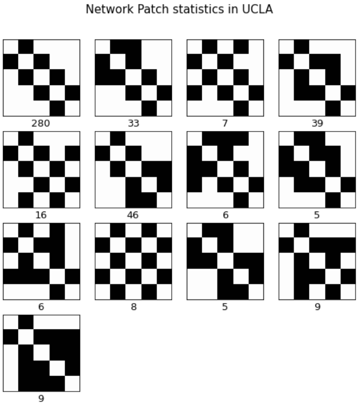
Statistics of k-node subgraph patterns





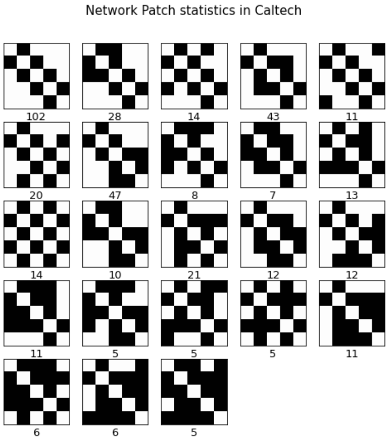
- What is the dimension of k-node connected subgraph patterns? (for large k)
- What are the essential subgraph patterns? (basis elements)

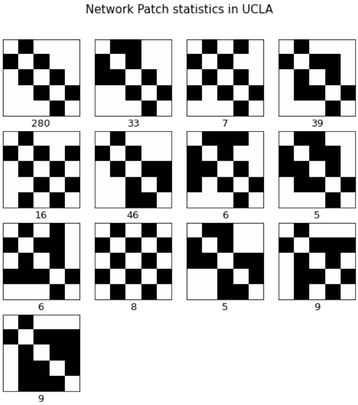




- What is the dimension of k-node connected subgraph patterns? (for large k)
- What are the essential subgraph patterns? (basis elements)
- How do they look like? (may depend on networks)

Statistics of k-node subgraph patterns



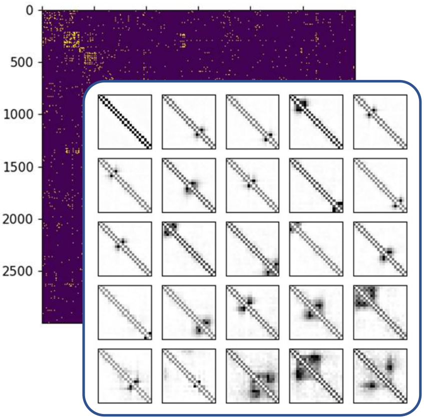


- What is the dimension of k-node connected subgraph patterns? (for large k)
- What are the essential subgraph patterns? (basis elements)
- How do they look like? (may depend on networks)

$\Rightarrow$ Algebraic properties of subgraph patterns

Network representation based on k-node connected subgraphs

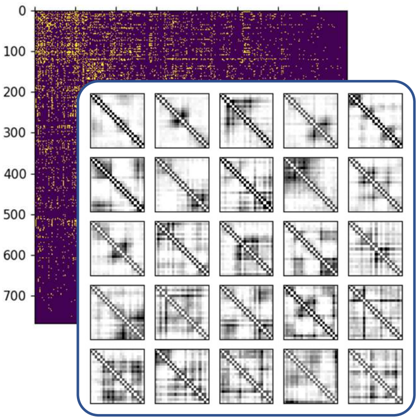
UCLA Facebook Network



b Network Dictionary

97% reconstruction accuracy

CALTECH Facebook Network

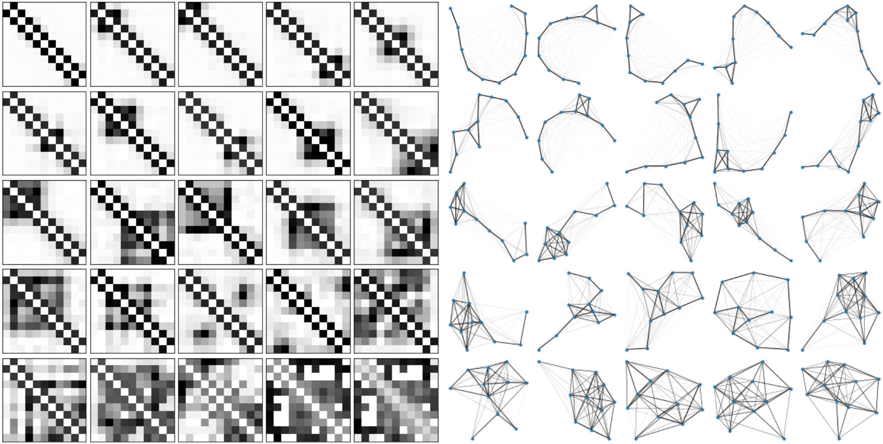


c Network Dictionary

82% reconstruction accuracy

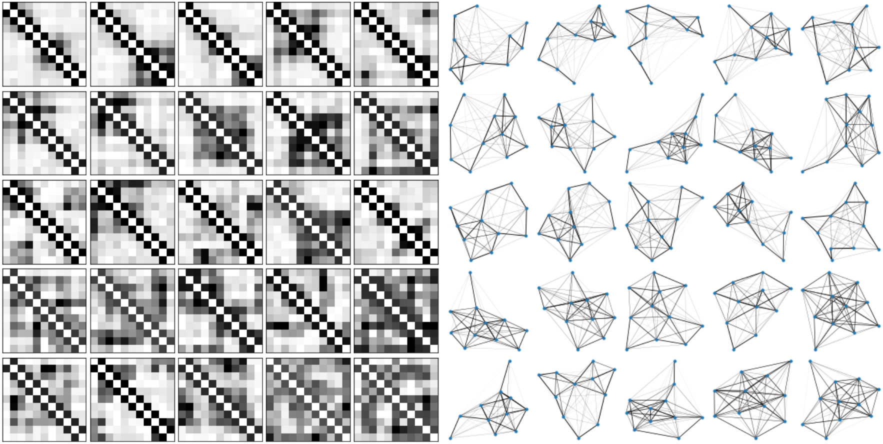
Network representation based on k-node connected subgraphs

Network Dictionary of UCLA ($k = 11$)



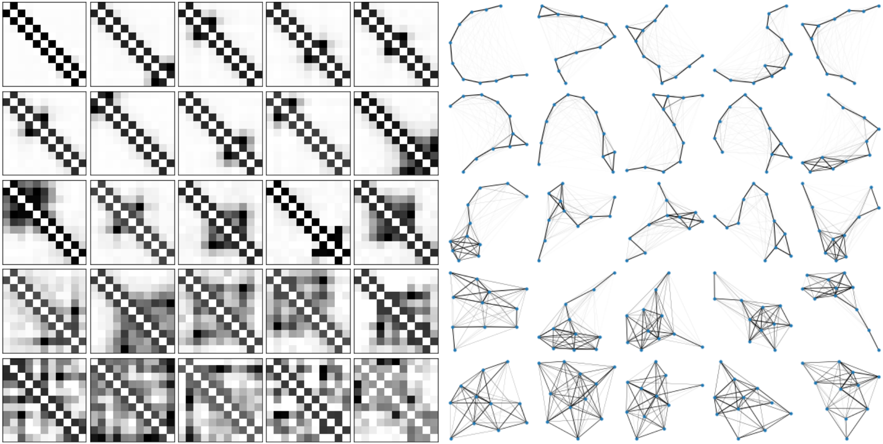
Network representation based on k-node connected subgraphs

Network Dictionary of Caltech ($k = 11$)

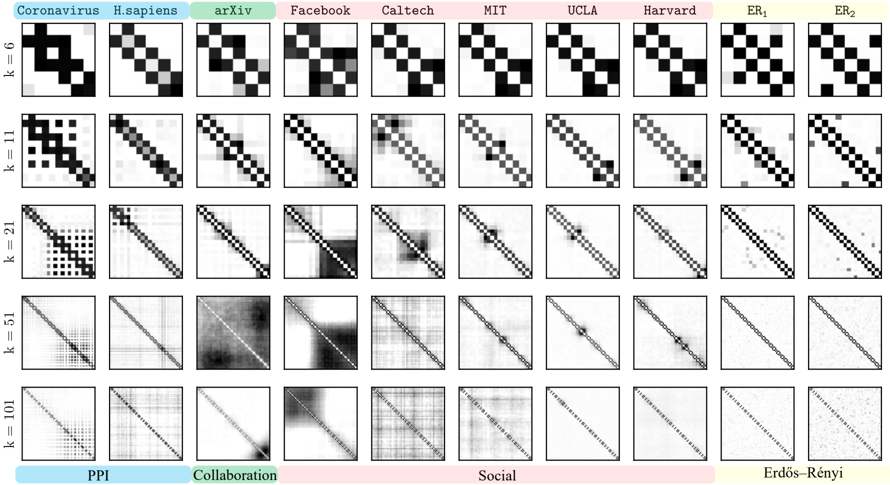


Network representation based on k-node connected subgraphs

Network Dictionary of Wisconsin ($k = 11$)



Network representation based on k-node connected subgraphs



Idea

- Goal: Give a network summary at intermediate levels

Idea

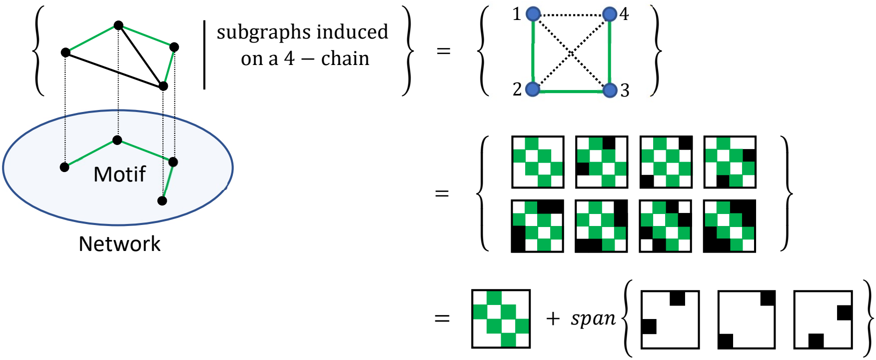
- Goal: Give a network summary at intermediate levels
  - Compute a large number of k-node connected subgraphs of a given network G

Idea

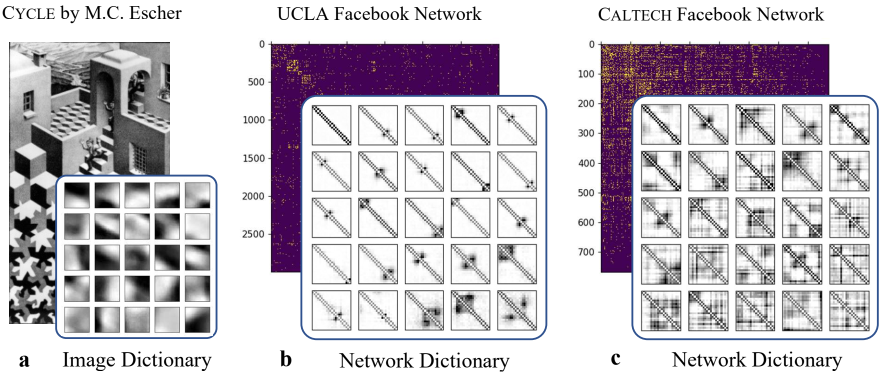
- Goal: Give a network summary at intermediate levels
  - Compute a large number of k-node connected subgraphs of a given network G
  - Find approximate non-negative basis for their adjacency matrices

Idea

- Goal: Give a network summary at intermediate levels
  - Compute a large number of *k*-node connected subgraphs of a given network *G*
  - Find approximate non-negative basis for their adjacency matrices

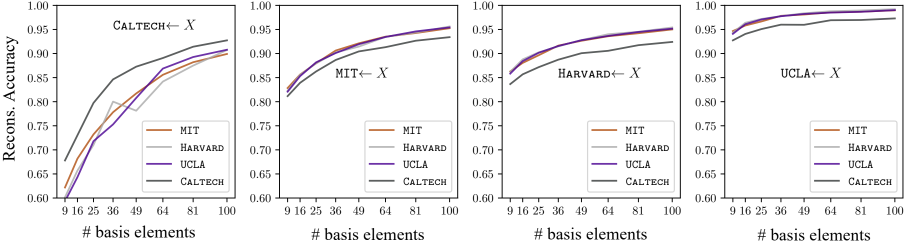


Network Dictionary Learning (NDL)



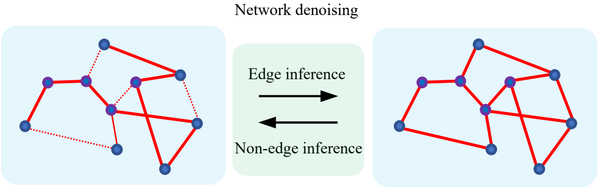
- NDL: Network data $\xrightarrow{\text{compress}}$ Latent motifs (nonnegative basis for subgraphs)
  - First introduced in L., Needell, Balzano [4]
  - Further developed in L., Kureh, Vendrow, Porter [5]

Applications of NDL to network reconstruction

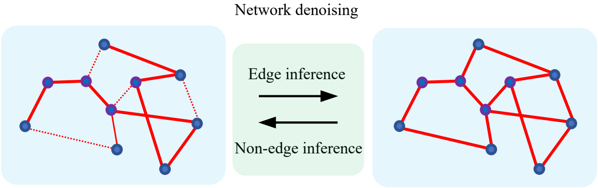


- $\text{Recons. Accuracy} = \dfrac{\#\text{ edges in original and recons.}}{\#\text{ edges in original or recons.}}$
- $k = 21$-node connected subgraphs
- Full dimension of the subgraph space $= 190$
- Many real-world netowrks have low-rank subgraph structures

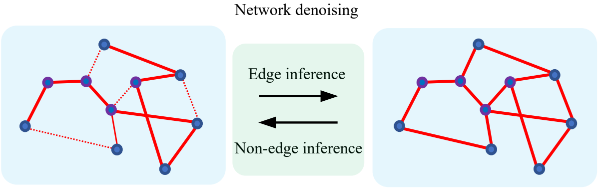
Applications to network denoising



Applications to network denoising



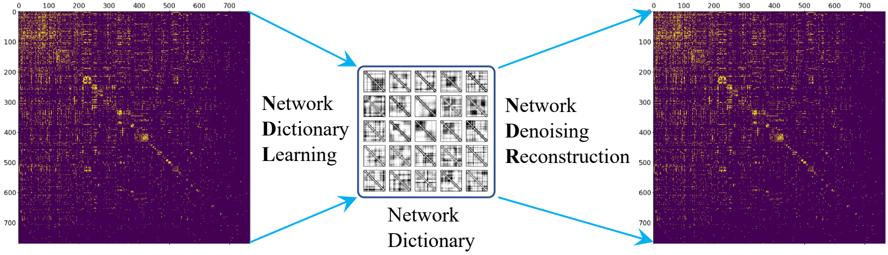
Applications: Recommender systems, community detection, anomaly detection, fraud detection



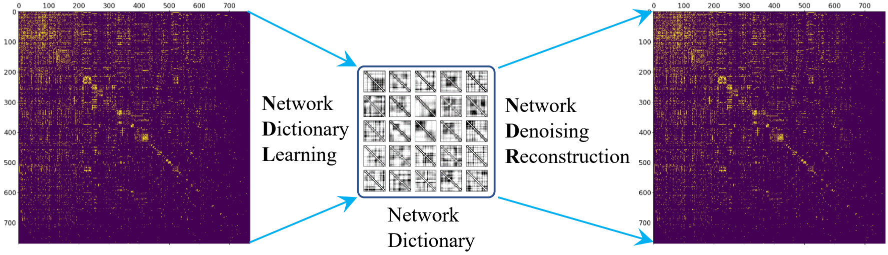
Applications: Recommender systems, community detection, anomaly detection, fraud detection

Algorithm		SNAP Facebook		H. sapiens		arXiv	
	Noise	+50%	-50%	+50%	-50%	+50%	-50%
Spec. Clustering		-	0.619	-	0.492	-	0.574
DeepWalk		-	0.968	-	0.744	-	0.934
LINE		-	0.949	-	0.725	-	0.890
node2vec		-	0.968	-	0.772	-	0.934
NDL+NDR		**0.979**	**0.981**	**0.814**	**0.859**	**0.950**	**0.954**

Contribution of our method in Network Data Analysis

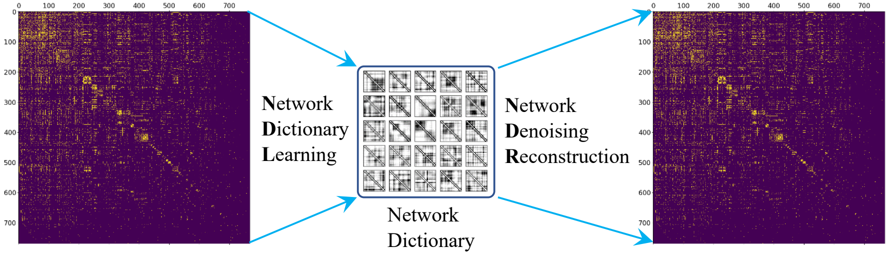


Contribution of our method in Network Data Analysis

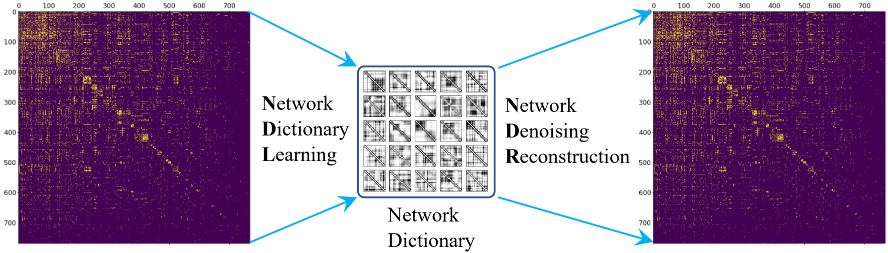


- Reveals network structure at intermediate scales

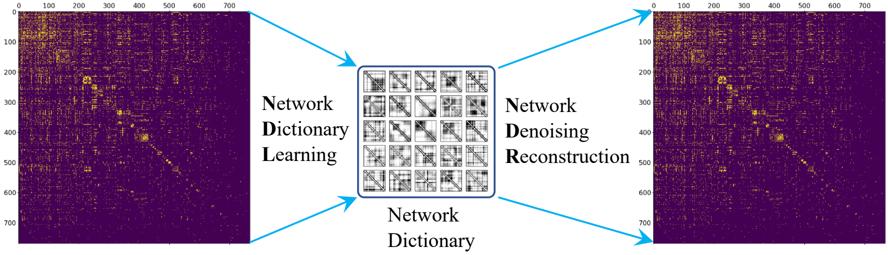
Contribution of our method in Network Data Analysis



- Reveals network structure at intermediate scales
  → Knowledge mining

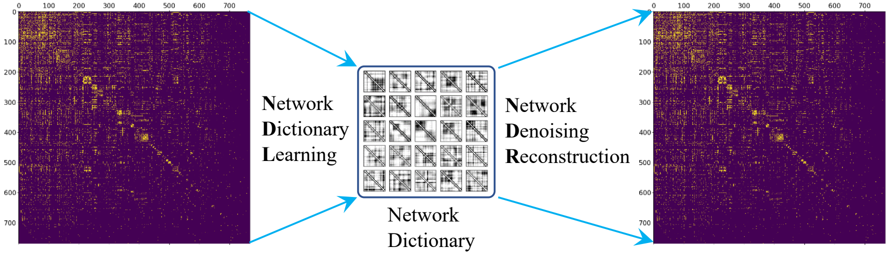


- Reveals network structure at intermediate scales
  $\rightarrow$ Knowledge mining
- Network data compression



- Reveals network structure at intermediate scales
  $\rightarrow$ Knowledge mining
- Network data compression
  $\rightarrow$ Clustering and classification for networks

Contribution of our method in Network Data Analysis



- Reveals network structure at intermediate scales

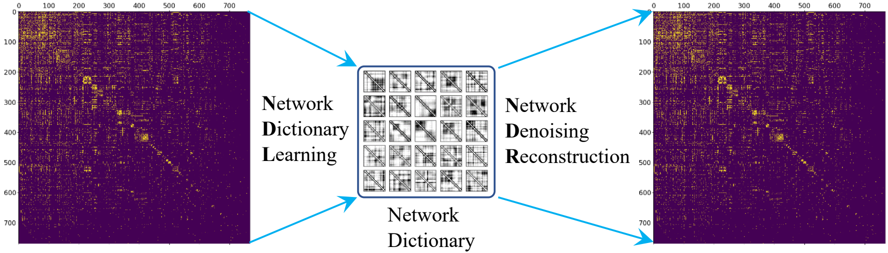
  $\rightarrow$ Knowledge mining

- Network data compression

  $\rightarrow$ Clustering and classification for networks

- Network denoising

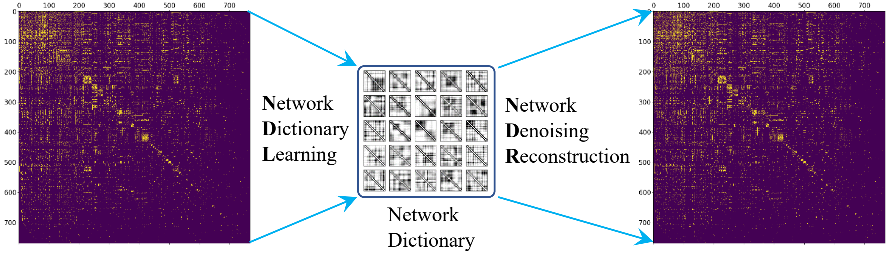
Contribution of our method in Network Data Analysis



- Reveals network structure at intermediate scales
  → Knowledge mining
- Network data compression
  → Clustering and classification for networks

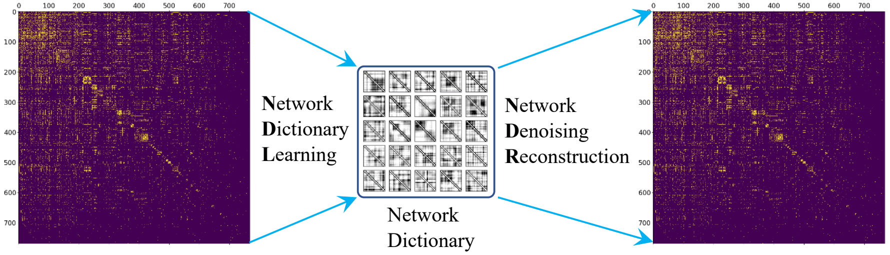
- Network denoising
  → Recommendation, faud detection

Contribution of our method in Network Data Analysis



- Reveals network structure at intermediate scales
  → Knowledge mining
- Network data compression
  → Clustering and classification for networks
- Network denoising
  → Recommendation, faud detection
- Transfer-reconstruction

Contribution of our method in Network Data Analysis



- Reveals network structure at intermediate scales
  $\rightarrow$ Knowledge mining
- Network data compression
  $\rightarrow$ Clustering and classification for networks
- Network denoising
  $\rightarrow$ Recommendation, faud detection
- Transfer-reconstruction
  $\rightarrow$ Network-level inference, disease association

CYCLE by M.C. Escher

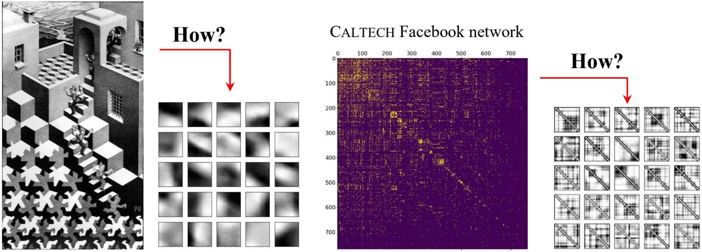


Image | Image Dictionary | Network | Network Dictionary

Main motivating question: *How do we learn dictionaries from images and networks?*

Outline

Dictionary Learning

Introduction to Network Dictionary Learning

Stochastic Optimization and Online Matrix Factorization

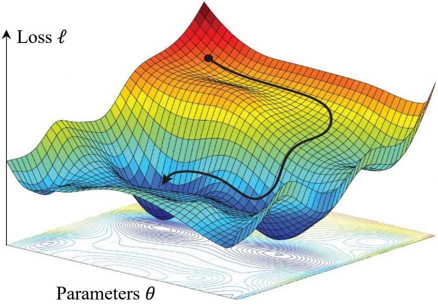
Theory and Main results

Proof ideas

Future directions and some ongoing works

What is Optimization?

- Optimization is a fundamental task whenever there is data to be explained by a model with parameters
- Data $\approx$ Model(θ)
  - e.g., Regression models (linear, logistic,..), latent variable models (matrix/tensor factorization,..), deep neural networks (CNN, RNN, GNN,..)



- How to chose optimal parameter θ^*?

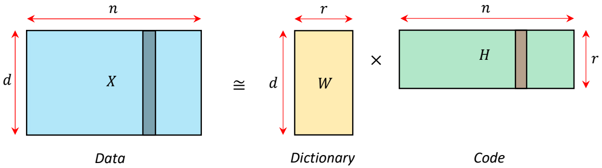
$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}} \ \ell(\text{Data}, \theta)$$

ℓ = Loss function

Θ = Parameter space

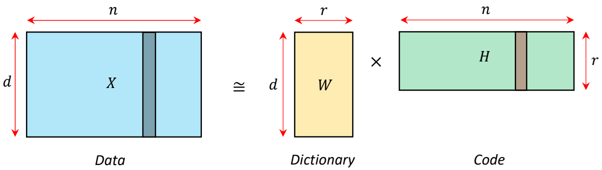
Matrix Factorization

- Matrix factorization is a fundamental tool in dictionary learning problems.



Matrix Factorization

- Matrix factorization is a fundamental tool in dictionary learning problems.

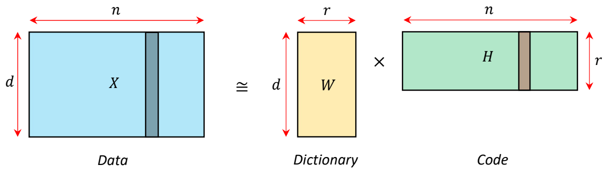


- Formulated as a non-convex optimization problem:

$$\begin{cases} \text{minimize} & \|X - WH\|_F^2 + \lambda \|H\|_1 \qquad \text{(Reconstruction error)} \\ \text{subject to} & W \in \mathcal{C},\ H \in \mathcal{C}' \qquad (\textit{Constraints}) \end{cases}$$

Matrix Factorization

- Matrix factorization is a fundamental tool in dictionary learning problems.

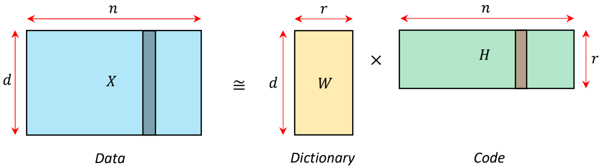


- Formulated as a non-convex optimization problem:

$$\begin{cases} \text{minimize} & \|X - WH\|_F^2 + \lambda \|H\|_1 & \text{(Reconstruction error)} \\ \text{subject to} & W \in \mathcal{C},\ H \in \mathcal{C}' & (\textit{Constraints}) \end{cases}$$

  - Nonnegative Matrix Factorization (NMF): $\mathcal{C} = \mathbb{R}_{\geq 0}^{d \times r}$, $\mathcal{C}' = \mathbb{R}_{\geq 0}^{r \times n}$, $\lambda = 0$,

- Matrix factorization is a fundamental tool in dictionary learning problems.



- Formulated as a non-convex optimization problem:

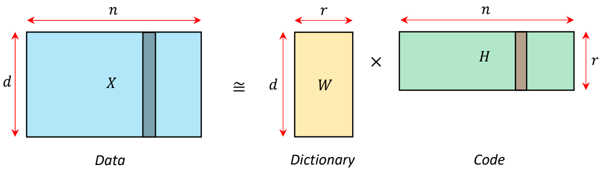
$$\begin{cases} \text{minimize} & \|X - WH\|_F^2 + \lambda \|H\|_1 \quad \text{(Reconstruction error)} \\ \text{subject to} & W \in \mathcal{C},\ H \in \mathcal{C}' \quad \textit{(Constraints)} \end{cases}$$

  - Nonnegative Matrix Factorization (NMF): $\mathcal{C} = \mathbb{R}_{\geq 0}^{d\times r}$, $\mathcal{C}' = \mathbb{R}_{\geq 0}^{r\times n}$, $\lambda = 0$,

Subspace clustering, Matrix Completion, Sparse PCA, Robust PCA, Poisson PCA, Heteroscedastic PCA,

Bilinear Inverse Problems, Max-Plus Factorization ...

- Matrix factorization is a fundamental tool in dictionary learning problems.



- Formulated as a non-convex optimization problem:

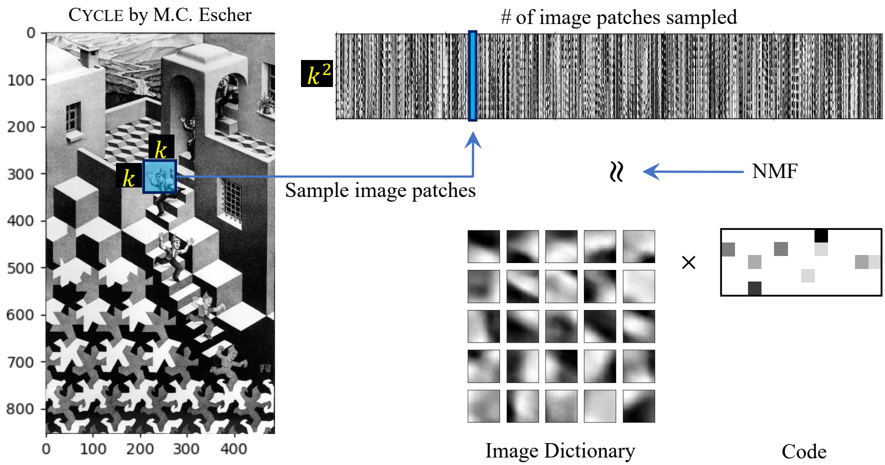
$$\begin{cases} \text{minimize} & \|X - WH\|_F^2 + \lambda \|H\|_1 \qquad \text{(Reconstruction error)} \\ \text{subject to} & W \in \mathcal{C},\ H \in \mathcal{C}' \qquad (\textit{Constraints}) \end{cases}$$

  - Nonnegative Matrix Factorization (NMF): $\mathcal{C} = \mathbb{R}_{\geq 0}^{d\times r}$, $\mathcal{C}' = \mathbb{R}_{\geq 0}^{r\times n}$, $\lambda = 0$,

    Subspace clustering, Matrix Completion, Sparse PCA, Robust PCA, Poisson PCA, Heteroscedastic PCA, Bilinear Inverse Problems, Max-Plus Factorization ...

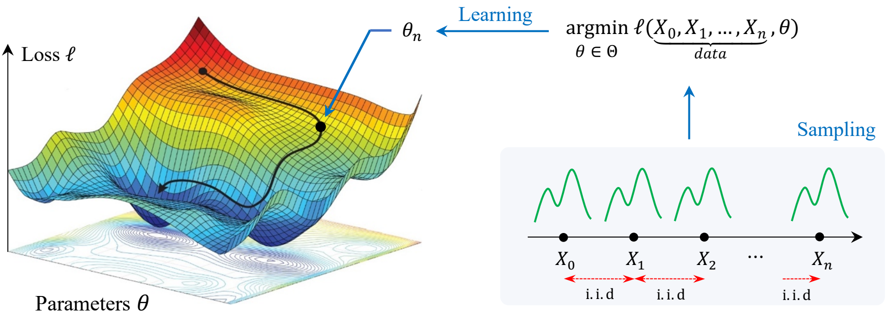
- Applications in text analysis, image reconstruction, medical imaging, bioinformatics, etc.

Example of NMF for Image dictionary learning



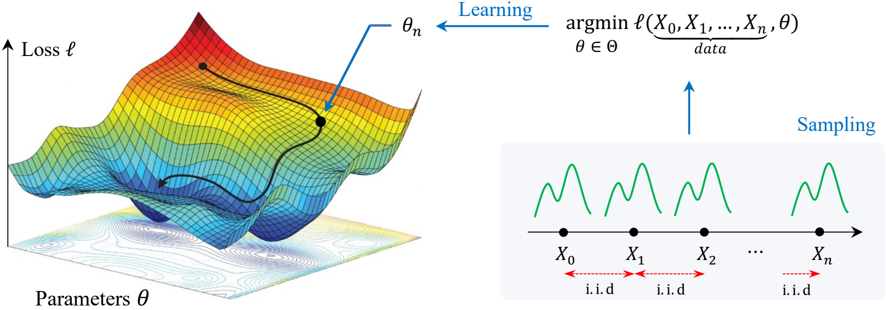
What is Stochastic Optimization?

- Stochastic optimization = optimization with random data samples



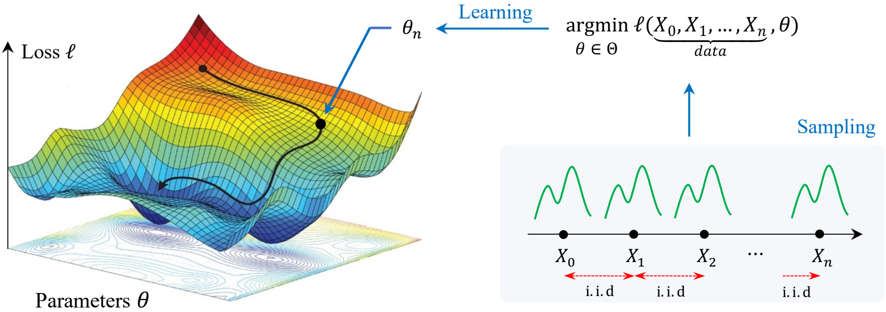
What is Stochastic Optimization?

- Stochastic optimization = optimization with random data samples
- Why use Stochastic Optimization?



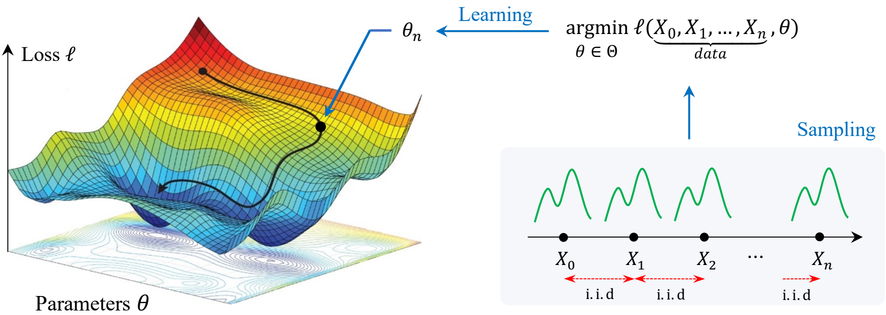
What is Stochastic Optimization?

- Stochastic optimization = optimization with random data samples
- Why use Stochastic Optimization?
  - The full data may not be available (yet to be observed, too large to process at once)



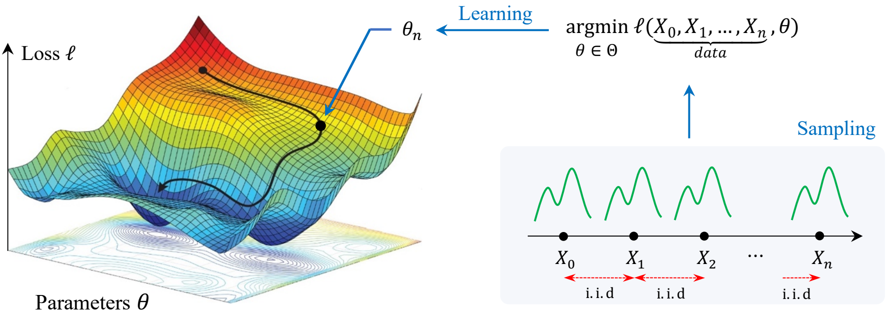
What is Stochastic Optimization?

- Stochastic optimization = optimization with random data samples
- Why use Stochastic Optimization?
  - The full data may not be available (yet to be observed, too large to process at once)
  - Sampling and optimization can be done simultaneously



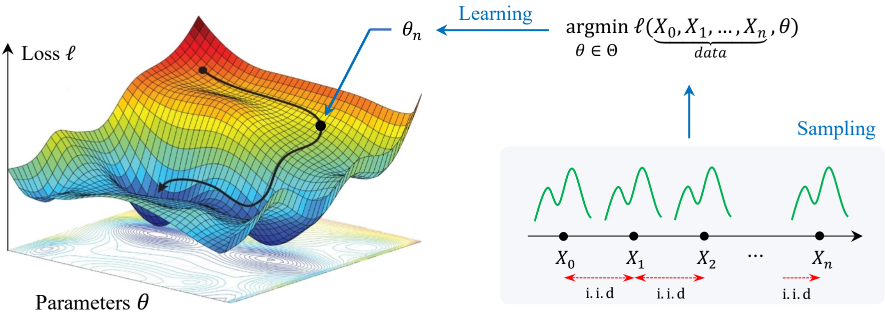
What is Stochastic Optimization?

- Many algorithms have been developed for i.i.d. data samples

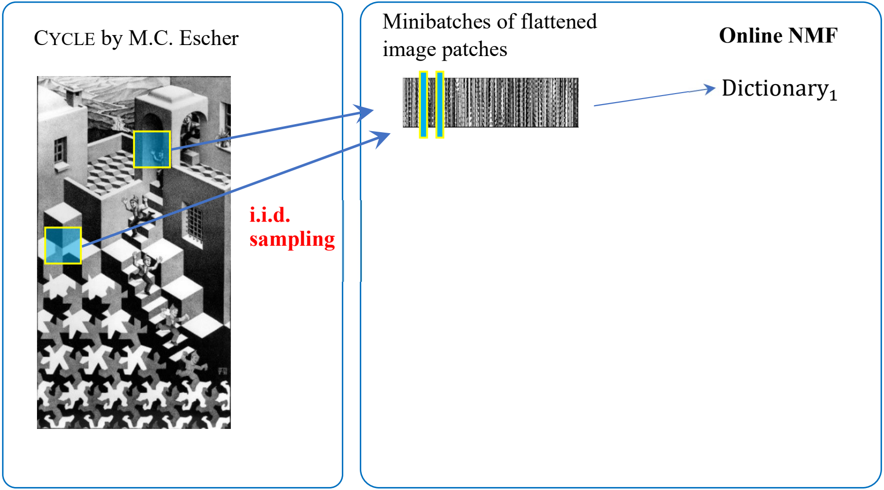


What is Stochastic Optimization?

- Many algorithms have been developed for i.i.d. data samples
  - e.g., Online (Stochastic) Matrix Factorization, Stochastic Gradient Descent, Stochastic Majorization-Minimization



Example of Online NMF + i.i.d. sampling: Image dictionary learning

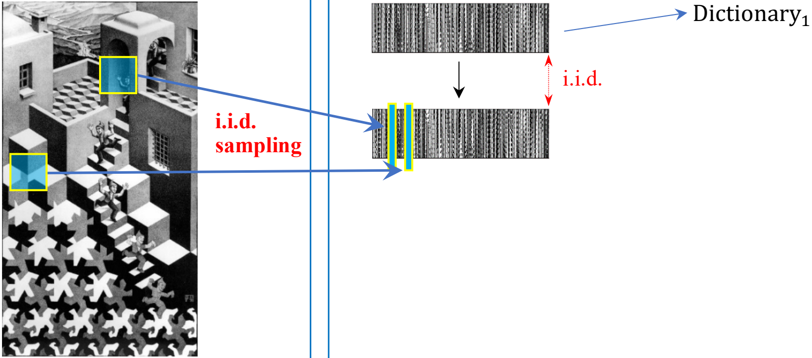


Example of Online NMF + i.i.d. sampling: Image dictionary learning

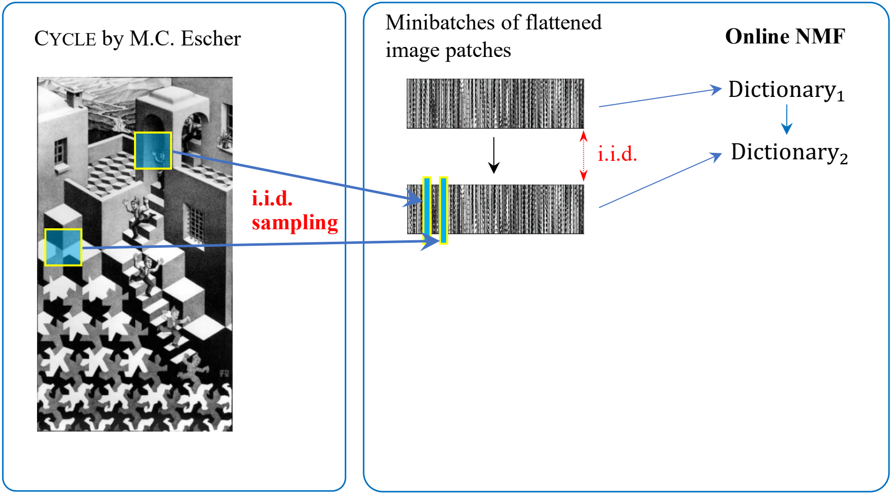
CYCLE by M.C. Escher

Minibatches of flattened image patches

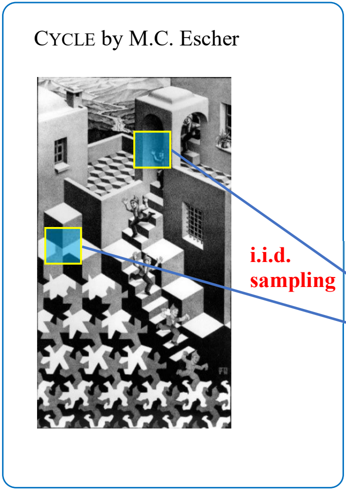
Online NMF

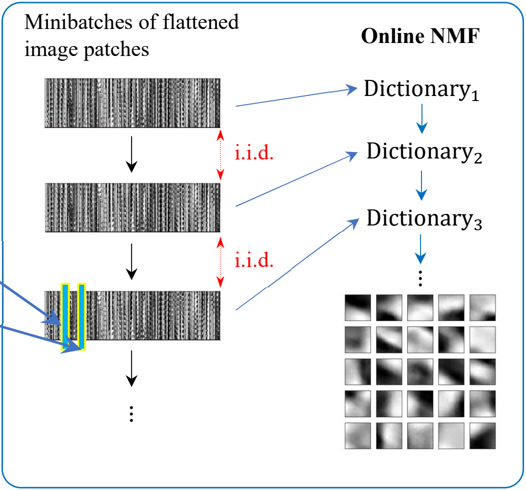


Example of Online NMF + i.i.d. sampling: Image dictionary learning

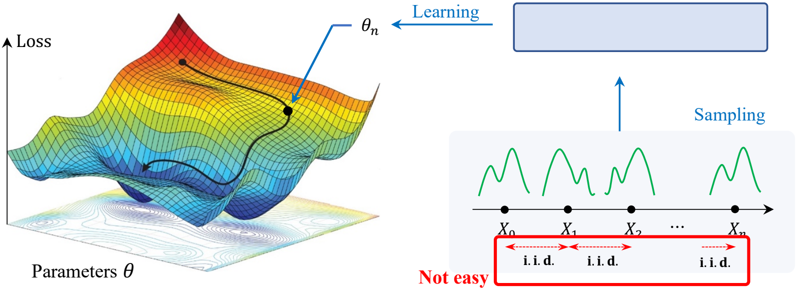


Example of Online NMF + i.i.d. sampling: Image dictionary learning

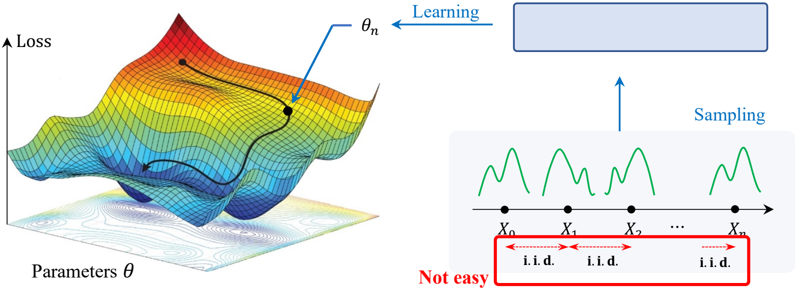




i.i.d. sampling is often expensive or infeasible



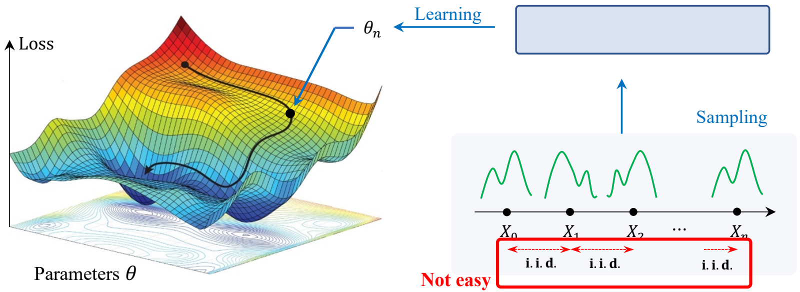
- However, i.i.d. sampling for many problems are difficult:



- However, i.i.d. sampling for many problems are difficult:

Posterior distribution

$$\pi(x) \propto \text{Likelihood}(\text{Data} \mid x)\ \text{prior}(x)$$



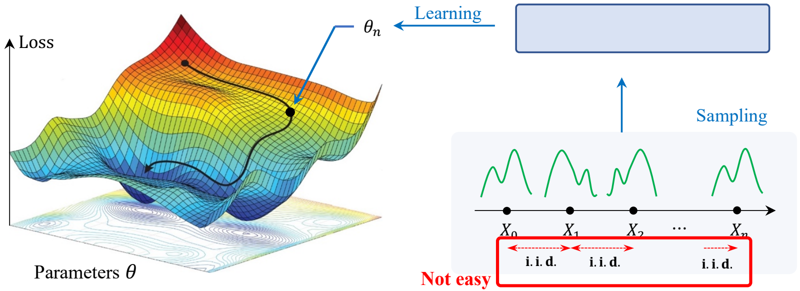
- However, i.i.d. sampling for many problems are difficult:

Posterior distribution

$$\pi(x) \propto \text{Likelihood}(\text{Data} \mid x)\ \text{prior}(x)$$

Gibbs measure (softmax dist.) (e.g., in Stat. physics, machine learning):

$$\pi(\text{face image } x) \propto \exp\left[0.2 * (\text{feature 1 of } x) + 0.7 * (\text{feature 2 of } x)\right]$$

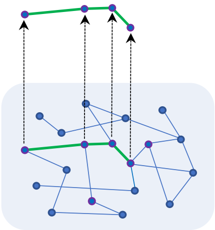


- However, i.i.d. sampling for many problems are difficult:

Motif sampling (Memoli, L., Sivakoff '19+ [3])
$F = ([k], E_F)$ motif, $G = (V, E)$ network. Sample $\mathbf{x} : [k] \to V$ from:

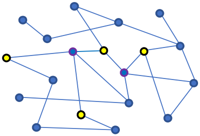
$$\pi(\mathbf{x}) \propto \mathbf{1}(\mathbf{x} : F \to G \text{ preserves all edges of } F)$$

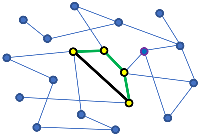
(Sample a graph homomorphism $F \to G$ uniformly)



Naive i.i.d. sampling can be infeasible

- Modern data (e.g., networks) are not only large, but also has intrinsic structure – could be lost by naive i.i.d. sampling

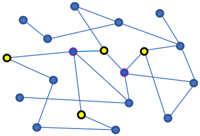


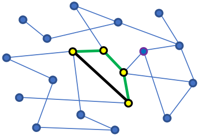


Stat	CORONAVIRUS	SNAP FB	ARXIV	CALTECH	MIT	UCLA	HARVARD
nodes	1555	4039	18772	769	6440	20467	15126
edges	4281	88234	198110	16656	251252	747613	824617
edge density	0.002	0.01	0.001	0.05	0.01	0.003	0.007

Naive i.i.d. sampling can be infeasible

- Modern data (e.g., networks) are not only large, but also has intrinsic structure – could be lost by naive i.i.d. sampling
  - Real-world networks are sparse $\rightarrow$ i.i.d. uniform sampling of k nodes returns almost no edges



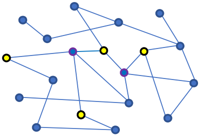


Stat	Coronavirus	SNAP FB	arXiv	Caltech	MIT	UCLA	Harvard
nodes	1555	4039	18772	769	6440	20467	15126
edges	4281	88234	198110	16656	251252	747613	824617
edge density	0.002	0.01	0.001	0.05	0.01	0.003	0.007

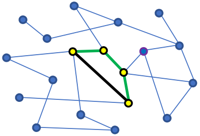
Naive i.i.d. sampling can be infeasible

- Modern data (e.g., networks) are not only large, but also has intrinsic structure – could be lost by naive i.i.d. sampling

  - Real-world networks are sparse → i.i.d. uniform sampling of k nodes returns almost no edges



  - Instead, sample a k-chain motif uniformly and take the induced subgraph — Motif sampling

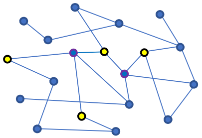


Stat	Coronavirus	SNAP FB	arXiv	Caltech	MIT	UCLA	Harvard
nodes	1555	4039	18772	769	6440	20467	15126
edges	4281	88234	198110	16656	251252	747613	824617
edge density	0.002	0.01	0.001	0.05	0.01	0.003	0.007

Naive i.i.d. sampling can be infeasible

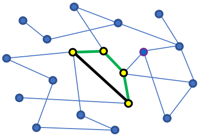
- Modern data (e.g., networks) are not only large, but also has intrinsic structure – could be lost by naive i.i.d. sampling

  - Real-world networks are sparse → i.i.d. uniform sampling of k nodes returns almost no edges



  - Instead, sample a k-chain motif uniformly and take the induced subgraph — Motif sampling

    — Additionally use rejection sampling to sample uniform Hamiltonian paths



Stat	Coronavirus	SNAP FB	arXiv	Caltech	MIT	UCLA	Harvard
nodes	1555	4039	18772	769	6440	20467	15126
edges	4281	88234	198110	16656	251252	747613	824617
edge density	0.002	0.01	0.001	0.05	0.01	0.003	0.007

Markov Chain Monte Carlo sampling

- Markov chain = Random walk on a sample space
    - (Future state | Current state, Past states) $\overset{d}{=}$ (Future state | Current state)

Markov Chain Monte Carlo sampling

- Markov chain = Random walk on a sample space
  - (Future state | Current state, Past states) $\overset{d}{=}$ (Future state | Current state)
- Markov Chain Monte Carlo (MCMC) sampling from π:

- Markov chain = Random walk on a sample space
  - (Future state | Current state, Past states) $\overset{d}{=}$ (Future state | Current state)

- Markov Chain Monte Carlo (MCMC) sampling from π:

(1) Design a Markov chain $(X_t)_{t\geq 0}$ such that $X_t \Rightarrow \pi$

- Markov chain = Random walk on a sample space
  - (Future state | Current state, Past states) $\overset{d}{=}$ (Future state | Current state)

- Markov Chain Monte Carlo (MCMC) sampling from π:

(1) Design a Markov chain $(X_t)_{t\geq 0}$ such that $X_t \Rightarrow \pi$

(2) Run the Markov chain for $T \gg 1$ iterations, then $X_T \sim \pi$ approximately

- Markov chain = Random walk on a sample space
  - (Future state | Current state, Past states) $\overset{d}{=}$ (Future state | Current state)

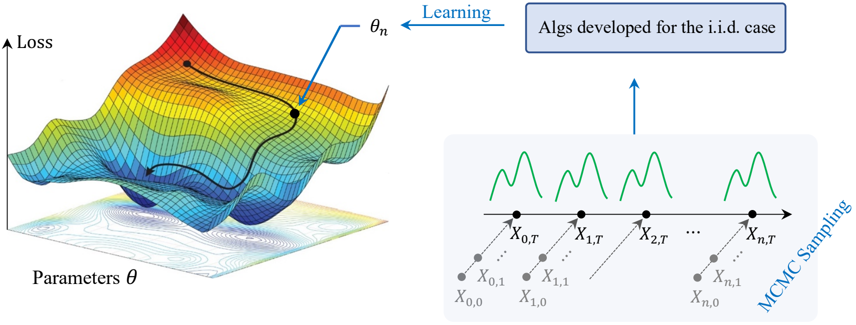
- Markov Chain Monte Carlo (MCMC) sampling from π:

(1) Design a Markov chain $(X_t)_{t\geq 0}$ such that $X_t \Rightarrow \pi$

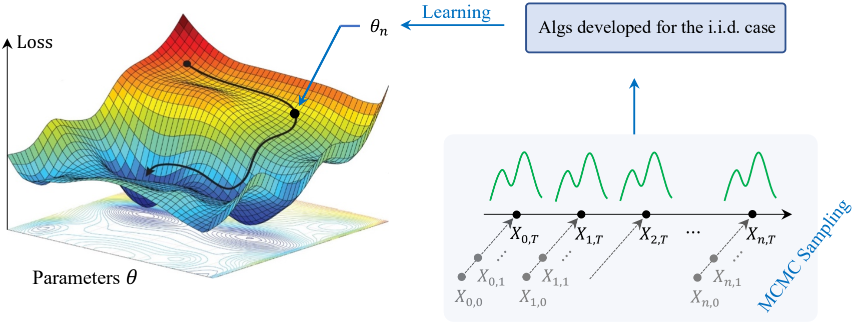
(2) Run the Markov chain for $T \gg 1$ iterations, then $X_T \sim \pi$ approximately

E.g. Random walk on graphs, PageRank, Gibbs sampling, Metropolis-Hastings algorithm, Langevin MC

- Standard approach:

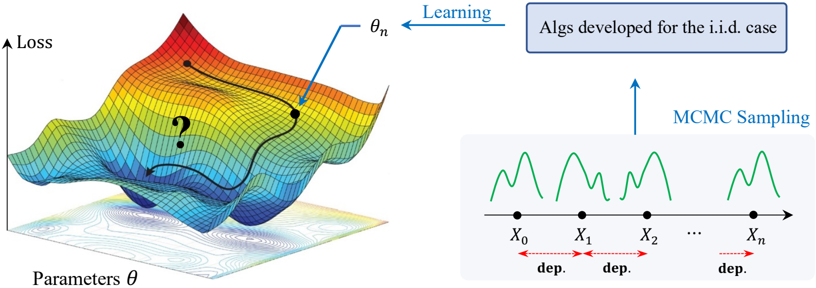


- Standard approach:



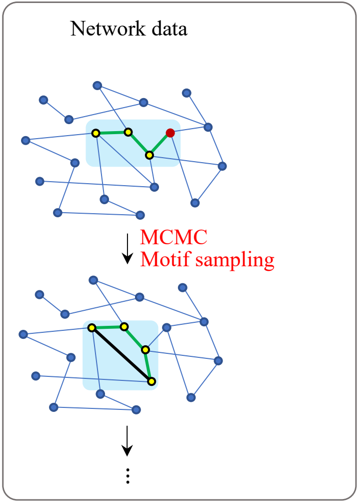
- Need to burn a MC for every single sample $\rightarrow$ Too many wasted samples

- Our approach: Optimize over a single MC trajectory

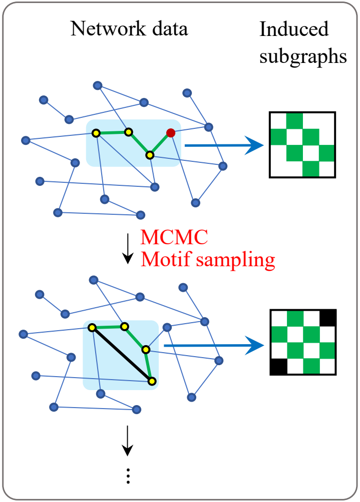


Network Dictionary Learning = Online NMF + MCMC motif sampling

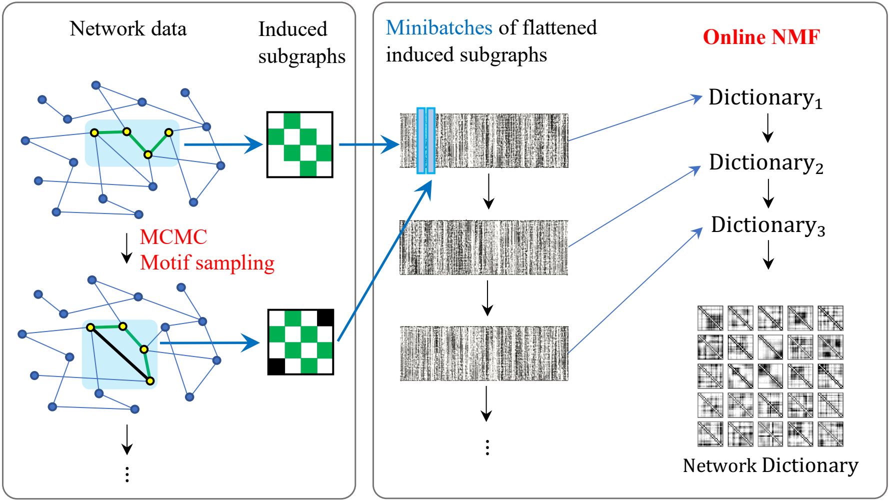
- MCMC motif sampling (Memoli, L., Sivakoff [3]): Uniformly samples a k-chain motif from network



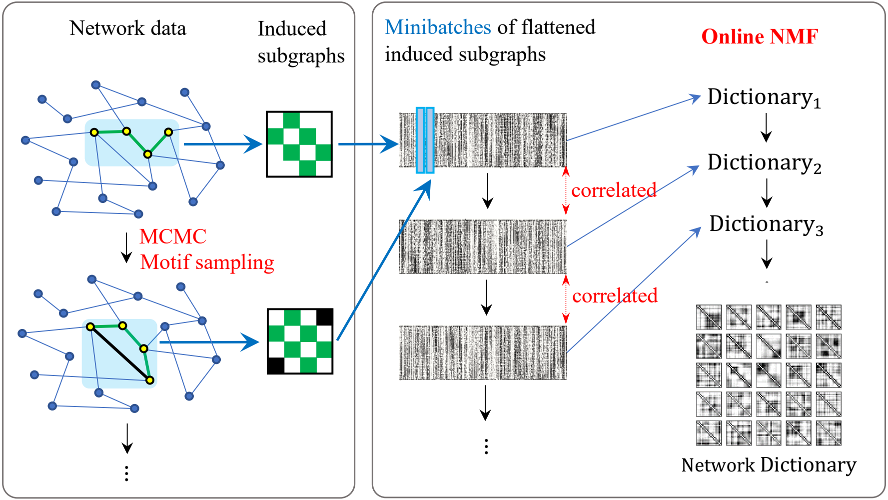
Network Dictionary Learning = Online NMF + MCMC motif sampling



Network Dictionary Learning = Online NMF + MCMC motif sampling



Network Dictionary Learning = Online NMF + MCMC motif sampling



Outline

Dictionary Learning

Introduction to Network Dictionary Learning

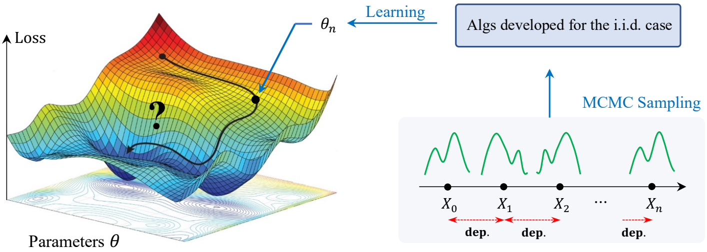
Stochastic Optimization and Online Matrix Factorization

Theory and Main results

Proof ideas

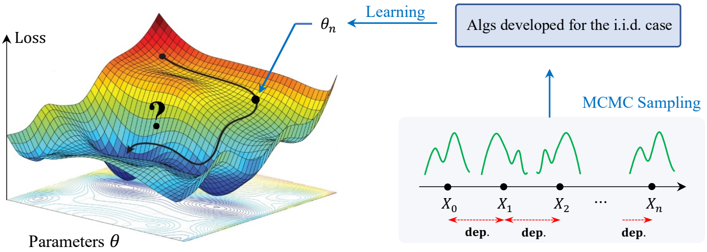
Future directions and some ongoing works

Stochastic Optimization + MCMC

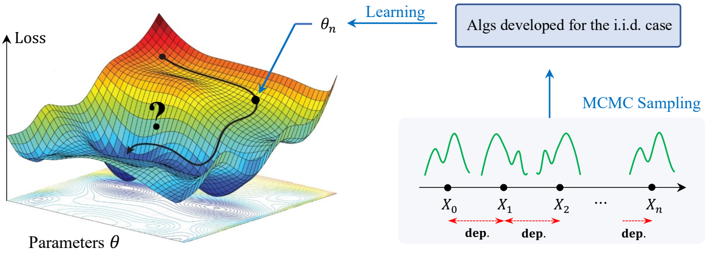


- **Question 1**: **Convergence to local min despite *data dependence*?**

Stochastic Optimization + MCMC

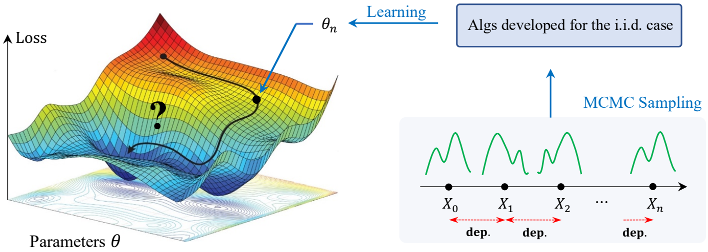


- **Question 1**: **Convergence to local min despite *data dependence*?**
  - **Main result 1.** We show a general convergence result with dependent data streams

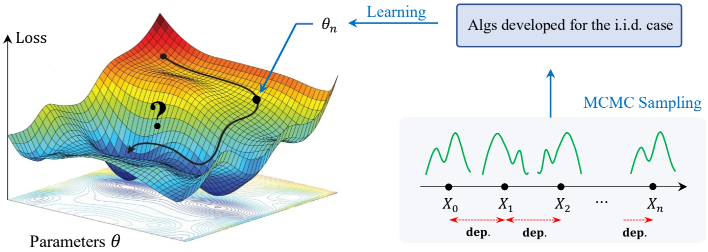


- **Question 1**: **Convergence to local min despite *data dependence*?**
  - **Main result 1.** We show a general convergence result with dependent data streams
    - Independence in data samples is crucial in convergence analysis [8, 6, 9]

Stochastic Optimization + MCMC



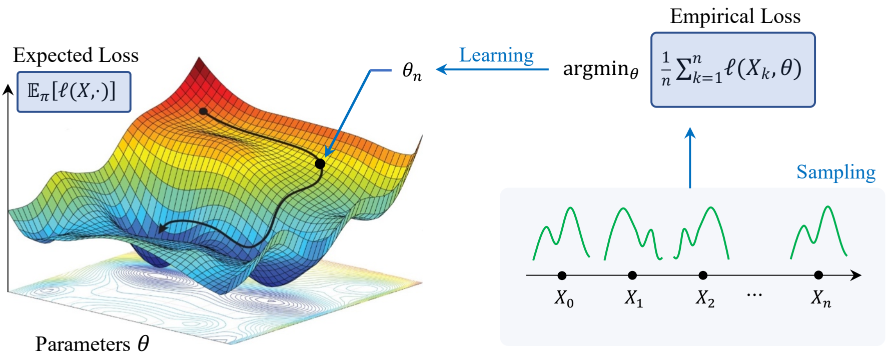
- **Question 1**: **Convergence to local min despite *data dependence*?**
  - **Main result 1.** We show a general convergence result with dependent data streams
    - Independence in data samples is crucial in convergence analysis [8, 6, 9]
- **Question 2**: **Rate of convergence and *data dependence*?**



- **Question 1**: **Convergence to local min despite *data dependence*?**
  - **Main result 1.** We show a general convergence result with dependent data streams
    - Independence in data samples is crucial in convergence analysis [8, 6, 9]
- **Question 2**: **Rate of convergence and *data dependence*?**
  - **Main result 2.** Rate of convergence = max(i.i.d. convergence rate, data correlation decay)

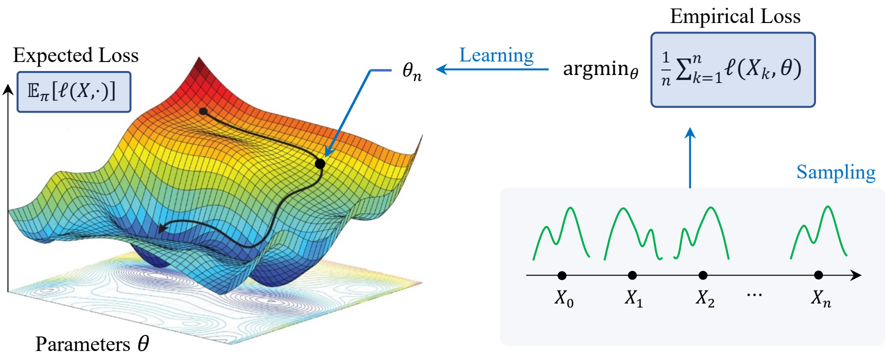
Empirical Loss Minimization

- Goal: Minimize the expected loss $\mathbb{E}_{X\sim\pi}[\ell(X,\theta)]$ given a loss function ℓ



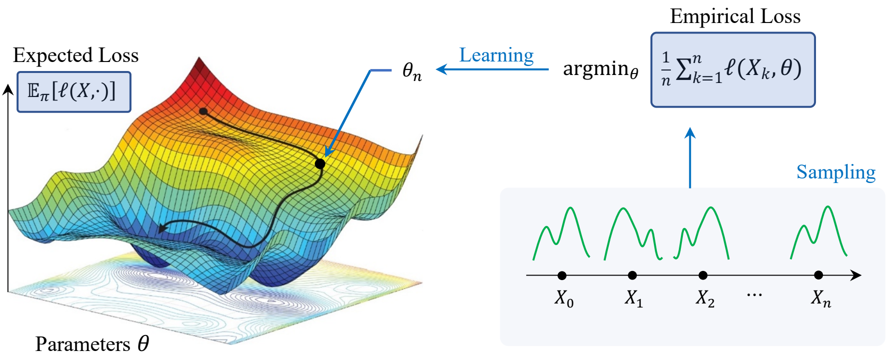
Empirical Loss Minimization

- Goal: Minimize the expected loss $\mathbb{E}_{X\sim\pi}[\ell(X,\theta)]$ given a loss function ℓ
- First attempt: *Empirical Loss Minimization*

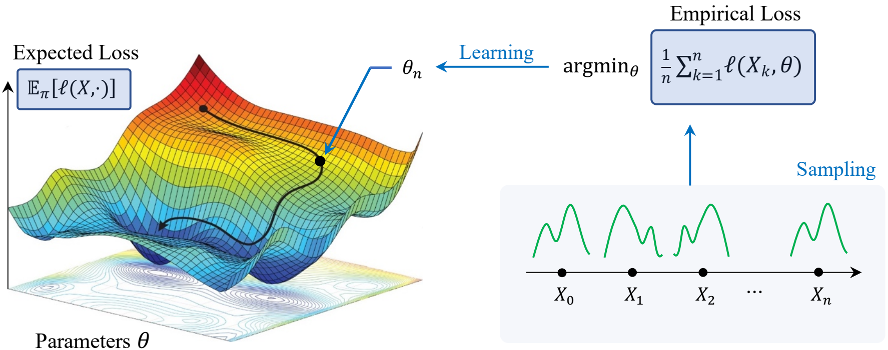


Empirical Loss Minimization

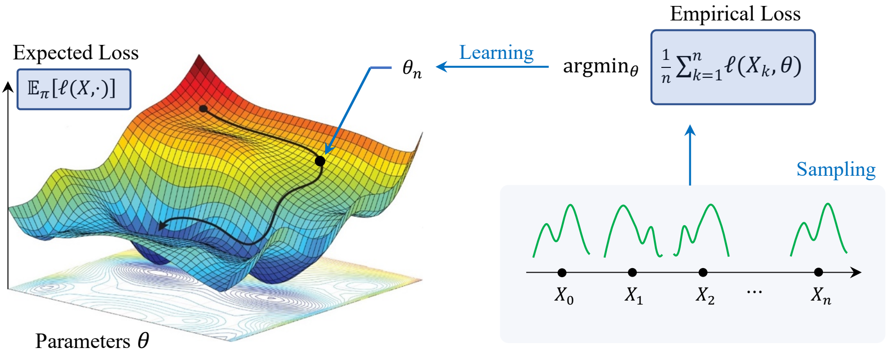
- Goal: Minimize the expected loss $\mathbb{E}_{X\sim\pi}[\ell(X,\theta)]$ given a loss function ℓ
- First attempt: *Empirical Loss Minimization*
  - Background: $\lim_{n\to\infty}$ Empirical Loss $=$ Expected loss



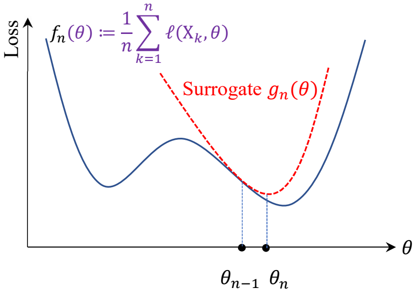
- Goal: Minimize the expected loss $\mathbb{E}_{X\sim\pi}[\ell(X,\theta)]$ given a loss function ℓ
- First attempt: *Empirical Loss Minimization*
  - Background: $\lim_{n\to\infty}$ Empirical Loss = Expected loss
  - Not practical in many cases:



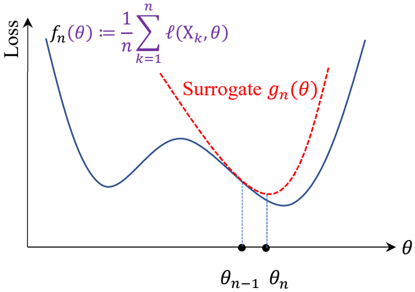
- Goal: Minimize the expected loss $\mathbb{E}_{X\sim\pi}[\ell(X,\theta)]$ given a loss function ℓ
- First attempt: *Empirical Loss Minimization*
  - Background: $\lim_{n\to\infty}$ Empirical Loss = Expected loss
  - Not practical in many cases:
    - The *empirical loss is often hard to minimize* (e.g., Matrix Factorization)



- Stochastic Majorization-Minimization (SMM) – Mairal [6]
  - Iteratively minimize majorizing surrogates g_n of the empirical loss f_n



- Stochastic Majorization-Minimization (SMM) – Mairal [6]
  - Iteratively minimize majorizing surrogates g_n of the empirical loss f_n



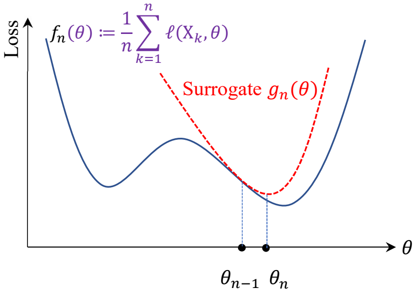
- Online Matrix Factorization in Mairal et al. [8]:

$$\text{(coding)} \quad H_n \leftarrow \operatorname*{argmin}_{H} \|X_n - \theta_{n-1} H\|_F^2$$

$$\text{(surrogate update)} \quad g_n(\theta) \leftarrow (1-w_n) g_{n-1}(\theta) + w_n \cdot \|X_n - \theta H_n\|_F^2$$

$$\text{(dictionary update)} \quad \theta_n \leftarrow \operatorname*{argmin}_{\theta \in \Theta} g_n(\theta)$$

- Stochastic Majorization-Minimization (SMM) – Mairal [6]
  - Iteratively minimize majorizing surrogates g_n of the empirical loss f_n



Online Tensor CP Factorization in Strohmeier, L., Needell et al. [11][10]:

$$\text{(coding)} \quad H_n \leftarrow \operatorname*{argmin}_{H} \|X_n - \mathrm{Out}(\theta_{n-1}, H)\|_F^2$$

$$\text{(surrogate update)} \quad g_n(\theta) \leftarrow (1 - w_n) g_{n-1}(\theta) + w_n \cdot \|X_n - \mathrm{Out}(\theta, H_n)\|_F^2$$

$$\text{(dictionary update)} \quad \theta_n \leftarrow \text{approx. } \operatorname*{argmin}_{\theta \in \Theta} g_n(\theta) \quad \text{(BCD with diminishing radius)}$$

Stochastic Majorization-Minimization

- SMM generalizes the Online Matrix Factorization algorithm in Mairal et al. [8]

- SMM generalizes the Online Matrix Factorization algorithm in Mairal et al. [8]
- When $\theta \mapsto \ell(X, \theta)$ is convex, $\theta_n \to$ global minimum at rate $O(\log n/\sqrt{n})$ for i.i.d. data samples X_n

- SMM generalizes the Online Matrix Factorization algorithm in Mairal et al. [8]
- When $\theta \mapsto \ell(X, \theta)$ is convex, $\theta_n \to$ global minimum at rate $O(\log n/\sqrt{n})$ for i.i.d. data samples X_n
- When $\theta \mapsto \ell(X, \theta)$ is non-convex, $\theta_n \to$ {local min of expected loss} for i.i.d. data samples X_n

Stochastic Majorization-Minimization

- SMM generalizes the Online Matrix Factorization algorithm in Mairal et al. [8]
- When $\theta \mapsto \ell(X, \theta)$ is convex, $\theta_n \to$ global minimum at rate $O(\log n/\sqrt{n})$ for i.i.d. data samples X_n
- When $\theta \mapsto \ell(X, \theta)$ is non-convex, $\theta_n \to \{$local min of expected loss$\}$ for i.i.d. data samples X_n (No known convergence rate)

Theorem (L. '20+ [2])

Suppose $\text{Data}_t = \text{function}(X_t)$, X_t *a Markov chain (irreducible, aperiodic, countable state) with* $\|\pi - \pi(X_n|X_{n-r})\|_{TV} = O(r^{-\gamma})$ *for some* $\gamma > 0$. $\boldsymbol{\theta}_n$:=*output of SMM.* $\Theta = \text{Set of constraints}$. *Under mild conditions,* Theorem (L. '20+ [2])

Suppose $\mathrm{Data}_t = \mathrm{function}(X_t)$, X_t *a Markov chain (irreducible, aperiodic, countable state) with* $\|\pi - \pi(X_n|X_{n-r})\|_{TV} = O(r^{-\gamma})$ *for some* $\gamma > 0$*.* $\boldsymbol{\theta}_n$ *:=output of SMM.* Θ = Set of constraints*. Under mild conditions,*

(i) $\boldsymbol{\theta}_n \to \{$*local min of expected loss over* $\Theta\}$ *a.s. as* $n \to \infty$*;*

Theorem (L. '20+ [2])

Suppose $\text{Data}_t = \text{function}(X_t)$, X_t *a Markov chain (irreducible, aperiodic, countable state) with* $\|\pi - \pi(X_n|X_{n-r})\|_{TV} = O(r^{-\gamma})$ *for some* $\gamma > 0$. $\boldsymbol{\theta}_n$ *:=output of SMM.* $\Theta = \text{Set of constraints}$. *Under mild conditions,*

(i) $\boldsymbol{\theta}_n \to \{$*local min of expected loss over* $\Theta\}$ *a.s. as* $n \to \infty$*;*

- *Implication: As long as the MC mixes polynomially fast, SMM estimates are locally consistent.*

Theorem (L. '20+ [2])

Suppose Data$_t$ = function(X_t), X_t *a Markov chain (irreducible, aperiodic, countable state) with* $\|\pi - \pi(X_n|X_{n-r})\|_{TV} = O(r^{-\gamma})$ *for some* $\gamma > 0$*.* $\boldsymbol{\theta}_n$ *:=output of SMM.* Θ = Set of constraints*. Under mild conditions,*

(i) $\boldsymbol{\theta}_n \to \{$*local min of expected loss over* $\Theta\}$ *a.s. as* $n \to \infty$*;*

- *Implication: As long as the MC mixes polynomially fast, SMM estimates are locally consistent.*
- *Note: Finite-state, irreducible, aperiodic MC* $\rightarrow$ *converges exponentially fast*

Convergence of SMM under dependent data streams —Non-convex case

Theorem (L. '20+ [2])

Suppose $\mathrm{Data}_t = \mathrm{function}(X_t)$, X_t *a Markov chain (irreducible, aperiodic, countable state) with* $\|\pi - \pi(X_n|X_{n-r})\|_{TV} = O(r^{-\gamma})$ *for some* $\gamma > 0$. $\boldsymbol{\theta}_n$ *:=output of SMM.* $\Theta =$ Set of constraints*. Under mild conditions,*

(i) $\boldsymbol{\theta}_n \to \{$*local min of expected loss over* $\Theta\}$ *a.s. as* $n \to \infty$*;*

- *Implication: As long as the MC mixes polynomially fast, SMM estimates are locally consistent.*
- *Note: Finite-state, irreducible, aperiodic MC* $\to$ *converges exponentially fast*

(ii) *Rate of convergence in gradients* $= O(\log n / n^{1/4})$*.*

Theorem (L. '20+ [2])

Suppose $\text{Data}_t = \text{function}(X_t)$, X_t *a Markov chain (irreducible, aperiodic, countable state) with* $\|\pi - \pi(X_n|X_{n-r})\|_{TV} = O(r^{-\gamma})$ *for some* $\gamma > 0$*.* $\boldsymbol{\theta}_n$ *:=output of SMM.* Θ = Set of constraints. *Under mild conditions,*

(i) $\boldsymbol{\theta}_n \to \{\textit{local min of expected loss over } \Theta\}$ *a.s. as* $n \to \infty$*;*

- *Implication: As long as the MC mixes polynomially fast, SMM estimates are locally consistent.*
- *Note: Finite-state, irreducible, aperiodic MC* $\to$ *converges exponentially fast*

(ii) *Rate of convergence in gradients* $= O(\log n / n^{1/4})$*.*

- *First rate of convergence result for constrained problem with dependent data*

Convergence of SMM under dependent data streams —Non-convex case

Theorem (L. '20+ [2])

Suppose $\text{Data}_t = \text{function}(X_t)$, X_t *a Markov chain (irreducible, aperiodic, countable state) with* $\|\pi - \pi(X_n|X_{n-r})\|_{TV} = O(r^{-\gamma})$ *for some* $\gamma > 0$. $\boldsymbol{\theta}_n$ *:=output of SMM.* Θ = Set of constraints. *Under mild conditions,*

(i) $\boldsymbol{\theta}_n \to \{$*local min of expected loss over* $\Theta\}$ *a.s. as* $n \to \infty$*;*

- *Implication: As long as the MC mixes polynomially fast, SMM estimates are locally consistent.*
- *Note: Finite-state, irreducible, aperiodic MC* $\rightarrow$ *converges exponentially fast*

(ii) *Rate of convergence in gradients* $= O(\log n/n^{1/4})$.

- *First rate of convergence result for constrained problem with dependent data*
- *For unconstrained problems, SGD is known to converge at rate* $O(\log n/n^{1/2})$

Theorem (L. '20+ [2])

Suppose $\text{Data}_t = \text{function}(X_t)$, X_t *a Markov chain (irreducible, aperiodic, countable state) with* $\|\pi - \pi(X_n|X_{n-r})\|_{TV} = O(r^{-\gamma})$ *for some* $\gamma > 0$*.* $\boldsymbol{\theta}_n$ *:=output of SMM.*
Θ = Set of constraints*. Under mild conditions,*

(i) $\boldsymbol{\theta}_n \to \{$*local min of expected loss over* $\Theta\}$ *a.s. as* $n \to \infty$*;*

- *Implication: As long as the MC mixes polynomially fast, SMM estimates are locally consistent.*
- *Note: Finite-state, irreducible, aperiodic MC* $\to$ *converges exponentially fast*

(ii) *Rate of convergence in gradients* $= O(\log n / n^{1/4})$*.*

- *First rate of convergence result for constrained problem with dependent data*
- *For unconstrained problems, SGD is known to converge at rate* $O(\log n / n^{1/2})$
- *Rate of* $O(n^{-1/4})$ *is afforded in order to handle constraints using MC CLT*

Theorem (L. '20+ [2])

Suppose $\text{Data}_t = \text{function}(X_t)$, X_t *a Markov chain (irreducible, aperiodic, countable state) with* $\|\pi - \pi(X_n|X_{n-r})\|_{TV} = O(r^{-\gamma})$ *for some* $\gamma > 0$. $\boldsymbol{\theta}_n$ *:=output of SMM.* Θ = Set of constraints. *Under mild conditions,*

(i) $\boldsymbol{\theta}_n \to \{$*local min of expected loss over* $\Theta\}$ *a.s. as* $n \to \infty$*;*

- *Implication: As long as the MC mixes polynomially fast, SMM estimates are locally consistent.*
- *Note: Finite-state, irreducible, aperiodic MC* $\to$ *converges exponentially fast*

(ii) *Rate of convergence in gradients* $= O(\log n / n^{1/4})$.

- *First rate of convergence result for constrained problem with dependent data*
- *For unconstrained problems, SGD is known to converge at rate* $O(\log n / n^{1/2})$
- *Rate of* $O(n^{-1/4})$ *is afforded in order to handle constraints using MC CLT*

Special cases: Online NMF (Mairal et al. '10 [8], L., Needell, Balzano '20 [4]), Online Nonnegativie Tensor CP-decomposition (Strohmeier, L., Needell '20 [11] [10])

Theorem (L., '20+ [2])

Suppose $f(\theta) = \mathbb{E}_\pi[\ell(\cdot, \theta)]$ *is convex.* π_n :=*distribution of* n^{th} *data point* X_n*,* $\bar{\theta}_n$:=*averaged output of SMM.* f :=*expected loss. Then for* $n \geq 1$*,*

$$\mathbb{E}\left[f(\bar{\theta}_{n-1}) - \min f\right] \leq \frac{c_1 + c_2 \sum_{k=1}^{n} k^{-1/2} \left(k^{-1/2} + \|\pi - \pi_k\|_{TV}\right)}{\sqrt{n}}.$$

Convergence of SMM under dependent data streams — Convex case

Theorem (L., '20+ [2])

Suppose $f(\theta) = \mathbb{E}_{\pi}[\ell(\cdot, \theta)]$ *is convex.* π_n :=*distribution of* n^{th} *data point* X_n*,* $\bar{\theta}_n$:=*averaged output of SMM.* f :=*expected loss. Then for* $n \geq 1$*,*

$$\mathbb{E}\left[f(\bar{\theta}_{n-1}) - \min f\right] \leq \frac{c_1 + c_2 \sum_{k=1}^{n} k^{-1/2}\left(k^{-1/2} + \|\pi - \pi_k\|_{TV}\right)}{\sqrt{n}}.$$

- No assumption on the structure of dependence in X_n

Theorem (L., '20+ [2])

Suppose $f(\theta) = \mathbb{E}_{\pi}[\ell(\cdot, \theta)]$ is convex. π_n :=distribution of n^{th} data point X_n, $\bar{\theta}_n$:=averaged output of SMM. f :=expected loss. Then for $n \geq 1$,

$$\mathbb{E}\left[f(\bar{\theta}_{n-1}) - \min f\right] \leq \frac{c_1 + c_2 \sum_{k=1}^{n} k^{-1/2} \left(k^{-1/2} + \|\pi - \pi_k\|_{TV}\right)}{\sqrt{n}}.$$

- No assumption on the structure of dependence in X_n

Corollary

Suppose f is convex.

(i) *If $\|\pi - \pi_n\|_{TV} = O(1/\sqrt{n})$, then $\mathbb{E}\left[f(\bar{\theta}_{n-1}) - \min f\right] = O(\log n/\sqrt{n})$.*

(ii) *If $\|\pi - \pi_n\|_{TV} \gg 1/\sqrt{n}$, then $\mathbb{E}\left[f(\bar{\theta}_{n-1}) - \min f\right] = \Theta(\|\pi - \pi_n\|_{TV})$.*

Theorem (L., '20+ [2])

Suppose $f(\theta) = \mathbb{E}_{\pi}[\ell(\cdot, \theta)]$ is convex. π_n :=distribution of n^{th} data point X_n, $\bar{\theta}_n$:=averaged output of SMM. f :=expected loss. Then for $n \geq 1$,

$$\mathbb{E}\left[f(\bar{\theta}_{n-1}) - \min f\right] \leq \frac{c_1 + c_2 \sum_{k=1}^{n} k^{-1/2} \left(k^{-1/2} + \|\pi - \pi_k\|_{TV}\right)}{\sqrt{n}}.$$

- No assumption on the structure of dependence in X_n

Corollary

Suppose f is convex.

(i) *If $\|\pi - \pi_n\|_{TV} = O(1/\sqrt{n})$, then $\mathbb{E}\left[f(\bar{\theta}_{n-1}) - \min f\right] = O(\log n/\sqrt{n})$.*

- *Implication: Better to use all data points without subsampling for case **(i)**;*

(ii) *If $\|\pi - \pi_n\|_{TV} \gg 1/\sqrt{n}$, then $\mathbb{E}\left[f(\bar{\theta}_{n-1}) - \min f\right] = \Theta(\|\pi - \pi_n\|_{TV})$.*

Theorem (L., '20+ [2])

Suppose $f(\theta) = \mathbb{E}_{\pi}[\ell(\cdot, \theta)]$ is convex. π_n :=distribution of n^{th} data point X_n, $\bar{\theta}_n$:=averaged output of SMM. f :=expected loss. Then for $n \geq 1$,

$$\mathbb{E}\left[f(\bar{\theta}_{n-1}) - \min f\right] \leq \frac{c_1 + c_2 \sum_{k=1}^{n} k^{-1/2} \left(k^{-1/2} + \|\pi - \pi_k\|_{TV}\right)}{\sqrt{n}}.$$

- No assumption on the structure of dependence in X_n

Corollary

Suppose f is convex.

(i) *If $\|\pi - \pi_n\|_{TV} = O(1/\sqrt{n})$, then $\mathbb{E}\left[f(\bar{\theta}_{n-1}) - \min f\right] = O(\log n/\sqrt{n})$.*

- *Implication: Better to use all data points without subsampling for case **(i)**;*

(ii) *If $\|\pi - \pi_n\|_{TV} \gg 1/\sqrt{n}$, then $\mathbb{E}\left[f(\bar{\theta}_{n-1}) - \min f\right] = \Theta(\|\pi - \pi_n\|_{TV})$.*

- *Implication 1: The trade-off between data dependence and information loss balances out exactly for convex problems*

Theorem (L., '20+ [2])

Suppose $f(\theta) = \mathbb{E}_\pi[\ell(\cdot, \theta)]$ is convex. π_n :=distribution of n^{th} data point X_n, $\bar{\theta}_n$:=averaged output of SMM. f :=expected loss. Then for $n \geq 1$,

$$\mathbb{E}\left[f(\bar{\theta}_{n-1}) - \min f\right] \leq \frac{c_1 + c_2 \sum_{k=1}^{n} k^{-1/2}\left(k^{-1/2} + \|\pi - \pi_k\|_{TV}\right)}{\sqrt{n}}.$$

- No assumption on the structure of dependence in X_n

Corollary

Suppose f is convex.

(i) *If $\|\pi - \pi_n\|_{TV} = O(1/\sqrt{n})$, then $\mathbb{E}\left[f(\bar{\theta}_{n-1}) - \min f\right] = O(\log n/\sqrt{n})$.*

- *Implication: Better to use all data points without subsampling for case **(i)**;*

(ii) *If $\|\pi - \pi_n\|_{TV} \gg 1/\sqrt{n}$, then $\mathbb{E}\left[f(\bar{\theta}_{n-1}) - \min f\right] = \Theta(\|\pi - \pi_n\|_{TV})$.*

- *Implication 1: The trade-off between data dependence and information loss balances out exactly for convex problems*
- *Implication 2: Subsampling does not improve convergence rate*

Theorem (L., '20+ [2])

Suppose $f(\theta) = \mathbb{E}_\pi[\ell(\cdot,\theta)]$ is convex. π_n :=distribution of n^{th} data point X_n, $\bar{\theta}_n$:=averaged output of SMM. f :=expected loss. Then for $n \geq 1$,

$$\mathbb{E}\left[f(\bar{\theta}_{n-1}) - \min f\right] \leq \frac{c_1 + c_2 \sum_{k=1}^n k^{-1/2}\left(k^{-1/2} + \|\pi - \pi_k\|_{TV}\right)}{\sqrt{n}}.$$

- No assumption on the structure of dependence in X_n

Corollary

Suppose f is convex.

(i) *If $\|\pi - \pi_n\|_{TV} = O(1/\sqrt{n})$, then $\mathbb{E}\left[f(\bar{\theta}_{n-1}) - \min f\right] = O(\log n/\sqrt{n})$.*

- *Implication: Better to use all data points without subsampling for case **(i)**;*

(ii) *If $\|\pi - \pi_n\|_{TV} \gg 1/\sqrt{n}$, then $\mathbb{E}\left[f(\bar{\theta}_{n-1}) - \min f\right] = \Theta(\|\pi - \pi_n\|_{TV})$.*

- *Implication 1: The trade-off between data dependence and information loss balances out exactly for convex problems*
- *Implication 2: Subsampling does not improve convergence rate*
- *We suspect this is not true for non-convex problems*

Outline

Two important lemmas to establish

- $$\Delta_n := \begin{cases} (\text{Relaxation error})_n := \overbrace{g_n(\theta_n)}^{\text{surrogate error at time } n} - \overbrace{f_n(\theta_n)}^{\text{empirical error at time } n} \geq 0 \\ (\text{Optimality gap})_n := \| \underbrace{\nabla g(\theta_n)}_{\perp \text{ to } \partial\Theta} - \nabla f(\theta_n) \|_F^2 \end{cases}$$

Two important lemmas to establish

- $$\Delta_n := \begin{cases} (\text{Relaxation error})_n := \overbrace{g_n(\theta_n)}^{\text{surrogate error at time } n} - \overbrace{f_n(\theta_n)}^{\text{empirical error at time } n} \geq 0 \\ (\text{Optimality gap})_n := \|\underbrace{\nabla g(\theta_n)}_{\perp \text{ to } \partial\Theta} - \nabla f(\theta_n)\|_F^2 \end{cases}$$

- **Lem 1**: $\sum_{n=0}^{\infty} w_n \, \mathbb{E}[\Delta_n] < \text{Abs. Const.} < \infty.$

Two important lemmas to establish

- $$\Delta_n := \begin{cases} (\text{Relaxation error})_n := \overbrace{g_n(\theta_n)}^{\text{surrogate error at time } n} - \overbrace{f_n(\theta_n)}^{\text{empirical error at time } n} \geq 0 \\ (\text{Optimality gap})_n := \|\underbrace{\nabla g(\theta_n)}_{\perp \text{ to } \partial\Theta} - \nabla f(\theta_n)\|_F^2 \end{cases}$$

- **Lem 1**: $\sum_{n=0}^{\infty} w_n \, \mathbb{E}[\Delta_n] < \text{Abs. Const.} < \infty$.
- **Lem 2**: $O\left(\mathbb{E}[\Delta_n] - \mathbb{E}[\Delta_{n-1}]\right) = O(w_n)$. $\cdots$ (not today:))

Two important lemmas to establish

- $$\Delta_n := \begin{cases} (\text{Relaxation error})_n := \overbrace{g_n(\theta_n)}^{\text{surrogate error at time } n} - \overbrace{f_n(\theta_n)}^{\text{empirical error at time } n} \geq 0 \\ (\text{Optimality gap})_n := \|\underbrace{\nabla g(\theta_n)}_{\perp \text{ to } \partial\Theta} - \nabla f(\theta_n)\|_F^2 \end{cases}$$
- **Lem 1**: $\sum_{n=0}^{\infty} w_n \, \mathbb{E}[\Delta_n] < \text{Abs. Const.} < \infty.$
- **Lem 2**: $O(\mathbb{E}[\Delta_n] - \mathbb{E}[\Delta_{n-1}]) = O(w_n).$ $\cdots$ (not today:))

  - From this, one can deduce

$$(1) \qquad \Delta_n \to 0 \quad \text{a.s. as } n \to \infty,$$

$$(2) \qquad \min_{1 \leq k \leq n} \sup_{\text{initialization}} \Delta_n = O\left(\frac{C}{\sum_{k=0}^{n} w_k} \right) \quad \text{a.a.s.}$$

First lemma

- After some nontrivial work, one can show

$$\sum_{n=0}^{\infty} w_{n+1} \mathbb{E}[\Delta_n] \leq c_1 + c_2 \sum_{n=0}^{\infty} w_{n+1} \left| \mathbb{E}\left[\underbrace{\ell(X_{n+1}, \theta_n)}_{\text{random loss at time } n+1} - \underbrace{f_n(\theta_n)}_{\text{empirical loss at time } n} \right] \right|$$

First lemma

- After some nontrivial work, one can show

$$\sum_{n=0}^{\infty} w_{n+1}\mathbb{E}[\Delta_n] \leq c_1 + c_2 \sum_{n=0}^{\infty} w_{n+1} \left| \mathbb{E}\left[\underbrace{\ell(X_{n+1},\theta_n)}_{\text{random loss at time } n+1} - \underbrace{f_n(\theta_n)}_{\text{empirical loss at time } n} \right] \right|$$

- Standard approach for the i.i.d. case:

First lemma

- After some nontrivial work, one can show

$$\sum_{n=0}^{\infty} w_{n+1}\mathbb{E}[\Delta_n] \le c_1 + c_2 \sum_{n=0}^{\infty} w_{n+1} \left| \mathbb{E}\left[\underbrace{\ell(X_{n+1}, \theta_n)}_{\text{random loss at time } n+1} - \underbrace{f_n(\theta_n)}_{\text{empirical loss at time } n} \right] \right|$$

- Standard approach for the i.i.d. case:

•
$$\begin{aligned} \mathbb{E}\left[\ell(X_{n+1}, \theta_n) - f_n(\theta_n)\right] &= \mathbb{E}\left[\mathbb{E}\left[\ell(X_{n+1}, \theta_n) - f_n(\theta_n) \,\middle|\, \mathcal{F}_n\right]\right] \\ &= \mathbb{E}\left[\underbrace{\mathbb{E}_{X\sim\pi}[\ell(X, \theta_n)] - f_n(\theta_n)}_{O(w_n\sqrt{n}) \text{ uniformly by uniform CLT}} \right] \end{aligned}$$

First lemma

- After some nontrivial work, one can show

$$\sum_{n=0}^{\infty} w_{n+1}\mathbb{E}[\Delta_n] \leq c_1 + c_2 \sum_{n=0}^{\infty} w_{n+1} \left| \mathbb{E}\left[\underbrace{\ell(X_{n+1}, \theta_n)}_{\text{random loss at time } n+1} - \underbrace{f_n(\theta_n)}_{\text{empirical loss at time } n} \right] \right|$$

- Standard approach for the i.i.d. case:
  - $$\mathbb{E}\left[\ell(X_{n+1}, \theta_n) - f_n(\theta_n)\right] = \mathbb{E}\left[\mathbb{E}\left[\ell(X_{n+1}, \theta_n) - f_n(\theta_n) \,\middle|\, \mathcal{F}_n\right]\right]$$
    $$= \mathbb{E}\left[\underbrace{\mathbb{E}_{X\sim\pi}[\ell(X, \theta_n)] - f_n(\theta_n)}_{O(w_n\sqrt{n}) \text{ uniformly by uniform CLT}} \right]$$
  - So the RHS above is $\leq C \sum_{n=1}^{\infty} w_n^2 \sqrt{n} < \infty$.

First lemma

- After some nontrivial work, one can show

$$\sum_{n=0}^{\infty} w_{n+1}\mathbb{E}[\Delta_n] \le c_1 + c_2 \sum_{n=0}^{\infty} w_{n+1} \left| \mathbb{E}\left[\underbrace{\ell(X_{n+1}, \theta_n)}_{\text{random loss at time } n+1} - \underbrace{f_n(\theta_n)}_{\text{empirical loss at time } n} \right] \right|$$

- Standard approach for the i.i.d. case:
  - $$\mathbb{E}\left[\ell(X_{n+1}, \theta_n) - f_n(\theta_n)\right] = \mathbb{E}\left[\mathbb{E}\left[\ell(X_{n+1}, \theta_n) - f_n(\theta_n) \,\middle|\, \mathcal{F}_n\right]\right]$$
    $$= \mathbb{E}\left[\underbrace{\mathbb{E}_{X\sim\pi}[\ell(X, \theta_n)] - f_n(\theta_n)}_{O(w_n\sqrt{n}) \text{ uniformly by uniform CLT}}\right]$$
  - So the RHS above is $\le C\sum_{n=1}^{\infty} w_n^2\sqrt{n} < \infty$.

  c.f.
  - $w_n \equiv$ stepsize in SGD
  - Nonconvex, unconstrained SGD convergence requires $\sum_{n=0}^{\infty} w_n^2 < \infty$
  - This is where we get $O(1/n^{1/4})$ SMM convergence instead of $O(1/n^{1/2})$ in SGD

First lemma

- After some nontrivial work, one can show

$$\sum_{n=0}^{\infty} w_{n+1}\mathbb{E}[\Delta_n] \leq c_1 + c_2 \sum_{n=0}^{\infty} w_{n+1} \left| \mathbb{E}\left[\underbrace{\ell(X_{n+1},\theta_n)}_{\text{random loss at time } n} - \underbrace{f_n(\theta_n)}_{\text{empirical loss at time } n} \right] \right|$$

- Our approach for the dependent case:

First lemma

- After some nontrivial work, one can show

$$\sum_{n=0}^{\infty} w_{n+1}\mathbb{E}[\Delta_n] \le c_1 + c_2 \sum_{n=0}^{\infty} w_{n+1} \left| \mathbb{E}\left[\underbrace{\ell(X_{n+1}, \theta_n)}_{\text{random loss at time } n} - \underbrace{f_n(\theta_n)}_{\text{empirical loss at time } n} \right] \right|$$

- Our approach for the dependent case:
  - Condition on distant past $\mathcal{F}_{n-\sqrt{n}}$ instead of the recent history $\mathcal{F}_n$:

$$\mathbb{E}\left[\ell(X_n, \theta_n) - f_n(\theta_n)\right] = \mathbb{E}\left[\mathbb{E}\left[\ell(X_{n+1}, \theta_n) - f_n(\theta_n) \,\middle|\, \mathcal{F}_{n-\sqrt{n}}\right]\right]$$

$$= \mathbb{E}\left[\underbrace{\mathbb{E}_{X\sim\pi}[\ell(X, \theta_n)] - f_{n-\sqrt{n}}(\theta_n)}_{O(w_n\sqrt{n}) \text{ uniformly by MC uniform CLT}} \right] + C \underbrace{\|\pi - \pi(\mathbf{x}_n | \mathcal{F}_{n-\sqrt{n}})\|_{TV}}_{\text{MC mixing: } O(\exp(-\sqrt{n}))}$$

First lemma

- After some nontrivial work, one can show

$$\sum_{n=0}^{\infty} w_{n+1}\mathbb{E}[\Delta_n] \le c_1 + c_2 \sum_{n=0}^{\infty} w_{n+1} \left| \mathbb{E}\left[\underbrace{\ell(X_{n+1},\theta_n)}_{\text{random loss at time } n} - \underbrace{f_n(\theta_n)}_{\text{empirical loss at time } n} \right] \right|$$

- Our approach for the dependent case:
  - Condition on distant past $\mathcal{F}_{n-\sqrt{n}}$ instead of the recent history $\mathcal{F}_n$:

$$\mathbb{E}\left[\ell(X_n,\theta_n) - f_n(\theta_n)\right] = \mathbb{E}\left[\mathbb{E}\left[\ell(X_{n+1},\theta_n) - f_n(\theta_n) \,\middle|\, \mathcal{F}_{n-\sqrt{n}}\right]\right]$$

$$= \mathbb{E}\left[\underbrace{\mathbb{E}_{X\sim\pi}[\ell(X,\theta_n)] - f_{n-\sqrt{n}}(\theta_n)}_{O(w_n\sqrt{n}) \text{ uniformly by MC uniform CLT}} \right] + C \underbrace{\|\pi - \pi(\mathbf{x}_n | \mathcal{F}_{n-\sqrt{n}})\|_{TV}}_{\text{MC mixing: } O(\exp(-\sqrt{n}))}$$

  - Again, the RHS above is $\le C' \sum_{n=1}^{\infty} w_n^2\sqrt{n} < \infty$.

- Supervised NDL and Network Regression
  $\longrightarrow$ Learn supervised subgraph patterns and regress

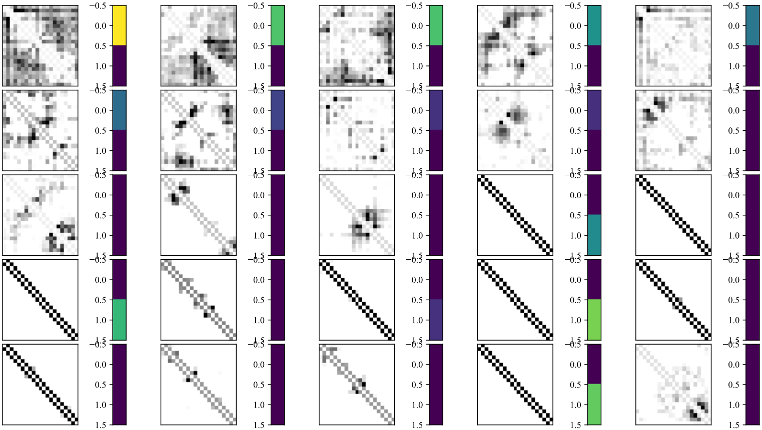


Figure: Supervised NDL between Caltech (label 0) and UCLA (label 1)

- Supervised NDL and Network Regression
  $\longrightarrow$ Learn supervised subgraph patterns and regress

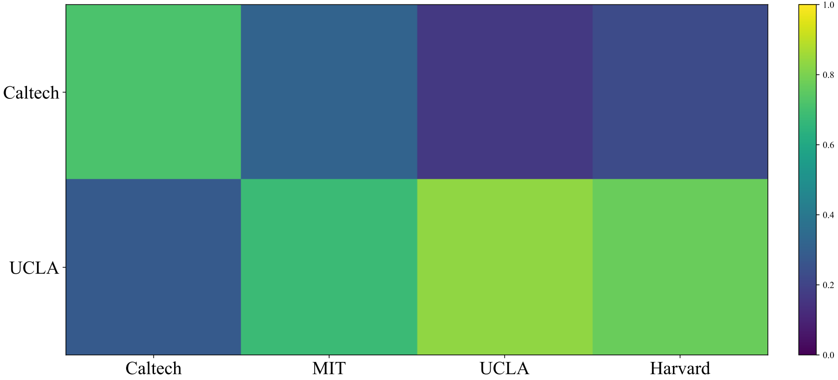


Figure: Network Regression

- Going from matrix factorizatino to tensor factorization
  $\longrightarrow$ Learn also from the time dimension

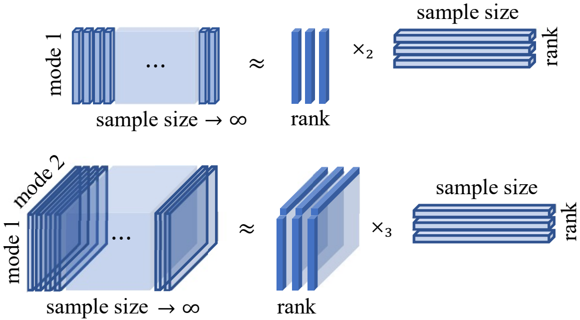


Figure: Online Matrix Factorization vs. Online Tensor Factorization

- Going from matrix factorization to tensor factorization
  $\longrightarrow$ Learn also from the time dimension

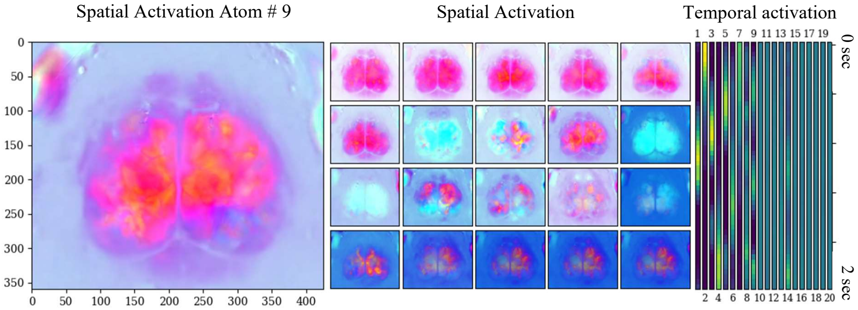


Figure: Temporal dictionary learned from mice brain activity video (Original data from Barson et al. *Nature methods* (2020))

- Online Tensor Factorization + Motif sampling $\longrightarrow$ NDL for Temporal Networks (Joint with Vendrow)

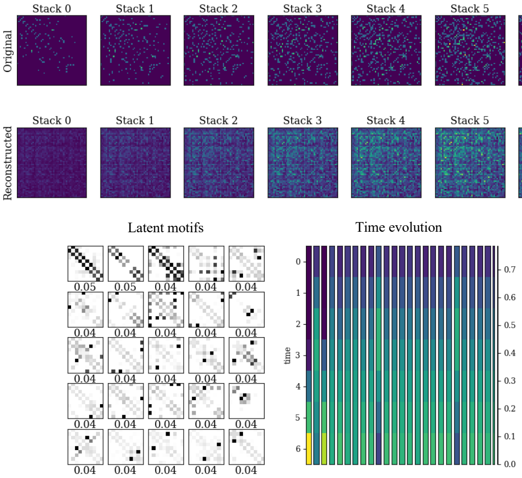


Figure: Temporal Network Dictionary learned from 7 stacks of 50 node graphs, 50 random edges added each time

Thanks!

[1] Michael Elad and Michal Aharon. "Image denoising via sparse and redundant representations over learned dictionaries". In: *IEEE Transactions on Image processing* 15.12 (2006), pp. 3736–3745.

[2] Hanbaek Lyu. "Alternating Stochastic Majorization-Minimization for dependent and multi-modal data streams". In: *In preperation* (2020).

[3] Hanbaek Lyu, Facundo Memoli, and David Sivakoff. "Sampling random graph homomorphisms and applications to network data analysis". In: *arXiv:1910.09483* (2019).

[4] Hanbaek Lyu, Deanna Needell, and Laura Balzano. "Online matrix factorization for Markovian data and applications to network dictionary learning". In: *Journal of Machine Learning Research 21 (to appear)* 21 (2021), pp. 1–49.

[5] Hanbaek Lyu et al. "Learning low-rank latent mesoscale structures in networks". In: *Draft available: https://hanbaeklyudotcom.files.wordpress.com/2020/10/ndl-1.pdf* (2020).

[6] Julien Mairal. "Stochastic majorization-minimization algorithms for large-scale optimization". In: *Advances in Neural Information Processing Systems*. 2013, pp. 2283–2291.

[7] Julien Mairal et al. "Non-local sparse models for image restoration". In: *2009 IEEE 12th international conference on computer vision*. IEEE. 2009, pp. 2272–2279.

[8] Julien Mairal et al. "Online learning for matrix factorization and sparse coding". In: *Journal of Machine Learning Research* 11 (2010), pp. 19–60.

[9] Arthur Mensch et al. "Stochastic subsampling for factorizing huge matrices". In: *IEEE Transactions on Signal Processing* 66.1 (2017), pp. 113–128.

[10] Christopher Strohmeier, Hanbaek Lyu, and Deanna Needell. "Online nonnegative CP tensor factorization and for Markovian data". In: *NeurIPS Workshop for Optimization for Machine Learning* (2020).

[11] Christopher Strohmeier, Hanbaek Lyu, and Deanna Needell. "Online nonnegative tensor factorization and CP-dictionary learning for Markovian data". In: *arXiv preprint arXiv:2009.07612* (2020).