# A Unifying Generative Model for Graph Learning Algorithms

Authors: Junteng Jia and Austin R. Benson
Presenter: Sudhanshu Chanpuriya

# Setting

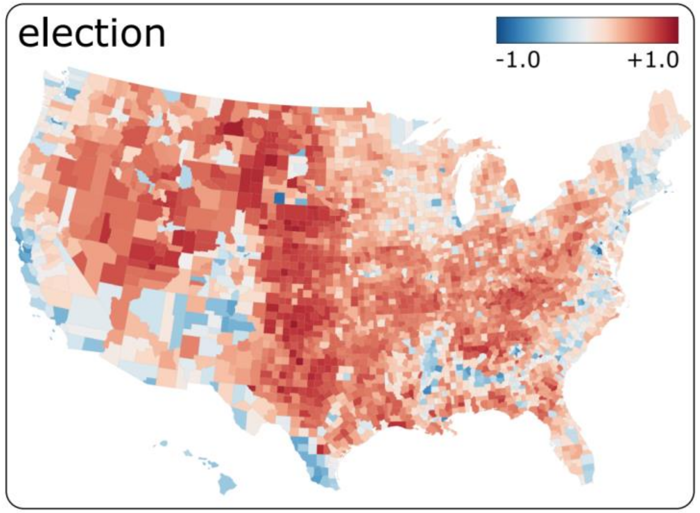
| | |
|---|---|
| **Graph** | $G = (V, E)$ |
| Weighted **adjacency** matrix | $\boldsymbol{W} \in \mathbb{R}^{n \times n}$ |
| **Degree** matrix | $\boldsymbol{D} = \mathrm{diag}(\mathbf{W1}) \in \mathbb{R}^{n \times n}$ |
| Symmetric **normed adj.** matrix | $\boldsymbol{S} = \boldsymbol{D}^{-1/2} \boldsymbol{W} \boldsymbol{D}^{-1/2} \in \mathbb{R}^{n \times n}$ |
| Normalized **Laplacian** | $\boldsymbol{N} = \boldsymbol{I} - \boldsymbol{S} \in \mathbb{R}^{n \times n}$ |

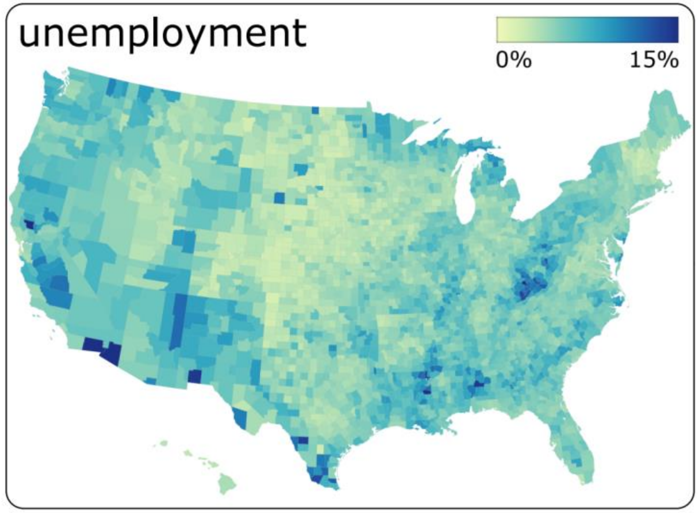
# Setting

| | |
|---|---|
| **Graph** | $G = (V, E)$ |
| Weighted **adjacency** matrix | $\boldsymbol{W} \in \mathbb{R}^{n \times n}$ |
| **Degree** matrix | $\boldsymbol{D} = \mathrm{diag}(\mathbf{W1}) \in \mathbb{R}^{n \times n}$ |
| Symmetric **normed adj.** matrix | $\boldsymbol{S} = \boldsymbol{D}^{-1/2} \boldsymbol{W} \boldsymbol{D}^{-1/2} \in \mathbb{R}^{n \times n}$ |
| Normalized **Laplacian** | $\boldsymbol{N} = \boldsymbol{I} - \boldsymbol{S} \in \mathbb{R}^{n \times n}$ |
| **Feature** matrix | $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ |
| **Label** vector | $\boldsymbol{y} \in \mathbb{R}^{n}$ |
| **Attribute** matrix | $\boldsymbol{A} = [\boldsymbol{X}\ \boldsymbol{y}] \in \mathbb{R}^{n \times (p+1)}$ |
| **Labeled** nodes | $L \in V$ |
| **Unlabeled** nodes | $U = V \backslash \mathrm{L} \in V$ |

# Setting

| | |
|---|---|
| **Graph** | $G = (V, E)$ |
| **Adjacency** | $\boldsymbol{W} \in \mathbb{R}^{n \times n}$ |
| **Degree** | $\boldsymbol{D} = \mathrm{diag}(\mathbf{W1}) \in \mathbb{R}^{n \times n}$ |
| **Normed adj.** | $\boldsymbol{S} = \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2} \in \mathbb{R}^{n \times n}$ |
| **Laplacian** | $\boldsymbol{N} = \boldsymbol{I} - \boldsymbol{S} \in \mathbb{R}^{n \times n}$ |
| **Feature** | $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ |
| **Label** | $\boldsymbol{y} \in \mathbb{R}^{n}$ |
| **Attribute** | $\boldsymbol{A} = [\boldsymbol{X}\ \boldsymbol{y}] \in \mathbb{R}^{n \times (p+1)}$ |
| **Labeled** | $L \in V$ |
| **Unlabeled** | $U = V \backslash \mathrm{L} \in V$ |



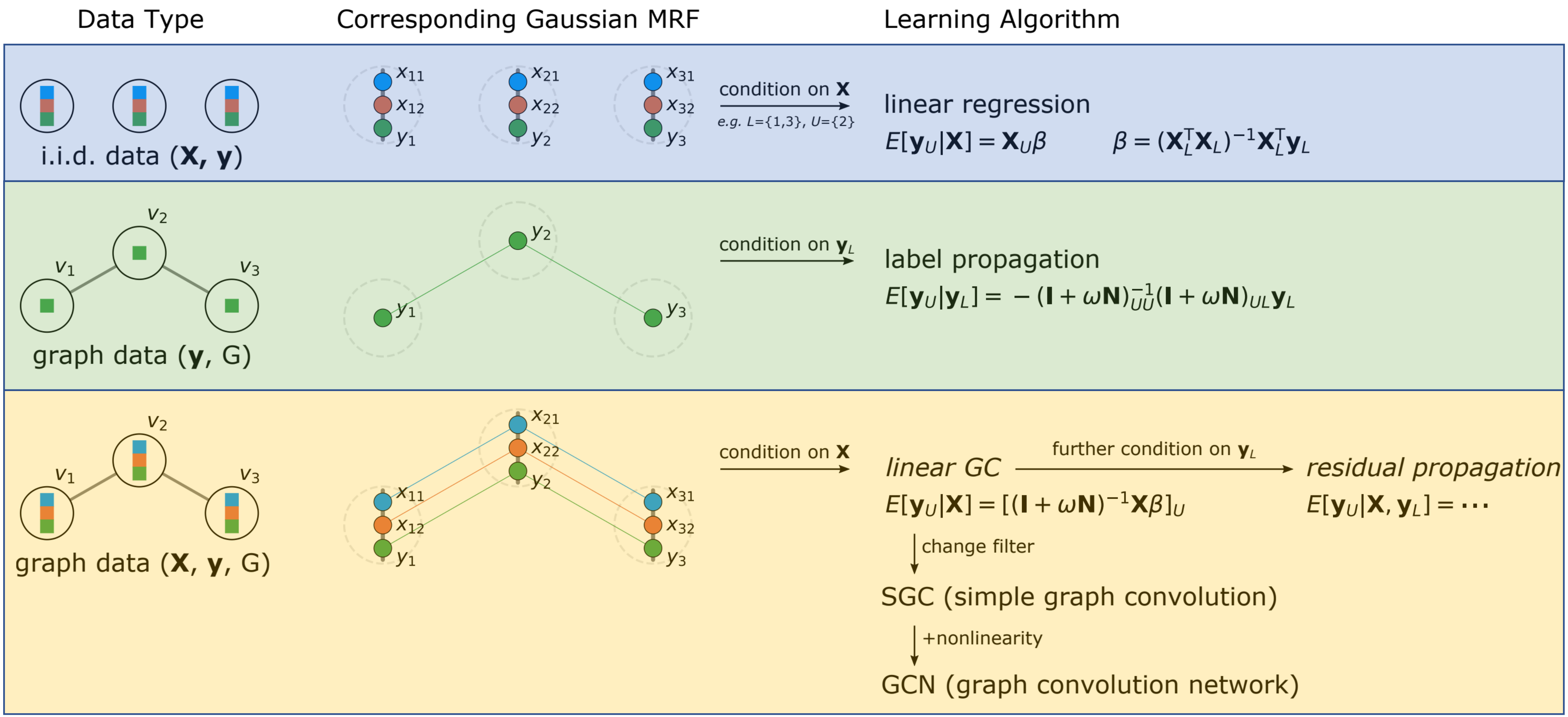




⋮

other attributes

# Setting

| | |
|---|---|
| **Graph** | $G = (V, E)$ |
| Weighted **adjacency** matrix | $\boldsymbol{W} \in \mathbb{R}^{n\times n}$ |
| **Degree** matrix | $\boldsymbol{D} = \mathrm{diag}(\mathbf{W1}) \in \mathbb{R}^{n\times n}$ |
| Symmetric **normed adj.** matrix | $\boldsymbol{S} = \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2} \in \mathbb{R}^{n\times n}$ |
| Normalized **Laplacian** | $\boldsymbol{N} = \boldsymbol{I} - \boldsymbol{S} \in \mathbb{R}^{n\times n}$ |
| **Feature** matrix (zero-centered) | $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ |
| **Label** vector (zero-centered) | $\boldsymbol{y} \in \mathbb{R}^{n}$ |
| **Attribute** matrix (zero-centered) | $\boldsymbol{A} = [\boldsymbol{X}\ \boldsymbol{y}] \in \mathbb{R}^{n\times(p+1)}$ |
| **Labeled** nodes | $L \in V$ |
| **Unlabeled** nodes | $U = V\backslash \mathrm{L} \in V$ |

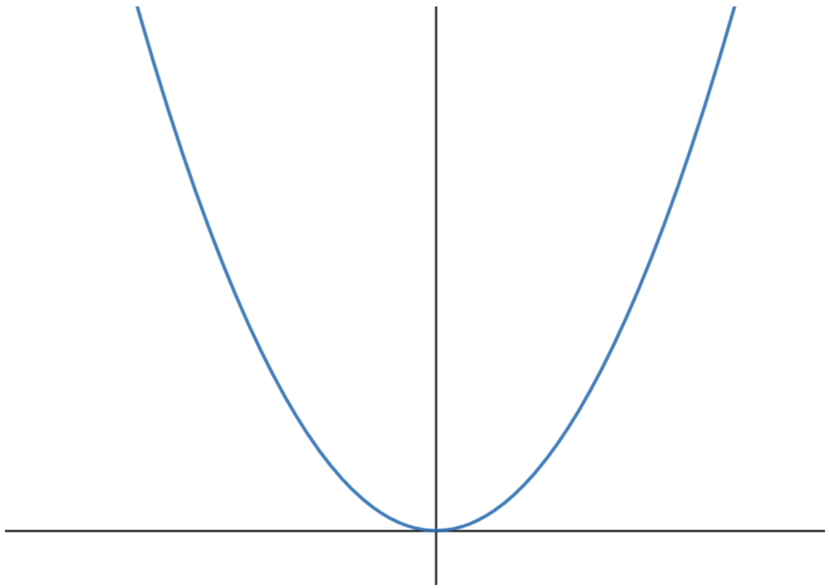


Data Type
Corresponding Gaussian MRF
Learning Algorithm
i.i.d. data (X, y)
x11 x12 y1 x21 x22 y2 x31 x32 y3
condition on X
e.g. L={1,3}, U={2}
linear regression
$E[\mathbf{y}_U|\mathbf{X}] = \mathbf{X}_U\beta$
$\beta = (\mathbf{X}_L^T\mathbf{X}_L)^{-1}\mathbf{X}_L^T\mathbf{y}_L$
graph data (y, G)
v1 v2 v3
condition on $\mathbf{y}_L$
label propagation
$E[\mathbf{y}_U|\mathbf{y}_L] = -(\mathbf{I}+\omega\mathbf{N})_{UU}^{-1}(\mathbf{I}+\omega\mathbf{N})_{UL}\mathbf{y}_L$
graph data (X, y, G)
condition on X
linear GC
further condition on $\mathbf{y}_L$
residual propagation
$E[\mathbf{y}_U|\mathbf{X}] = [(\mathbf{I}+\omega\mathbf{N})^{-1}\mathbf{X}\beta]_U$
$E[\mathbf{y}_U|\mathbf{X},\mathbf{y}_L] = \cdots$
change filter
SGC (simple graph convolution)
+nonlinearity
GCN (graph convolution network)

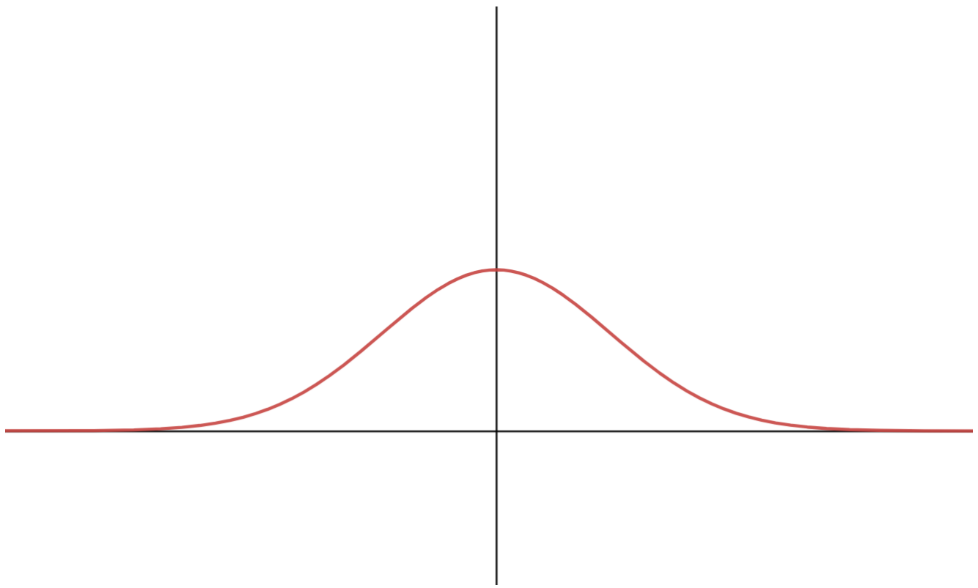
# Outline

1. Prerequisites
   a. Normal distributions
   b. Marginalizing and conditioning normals
2. Unifying model
3. Deriving learning algorithms
   a. Linear regression
   b. Label propagation
   c. Graph convolution

# Normal Distribution

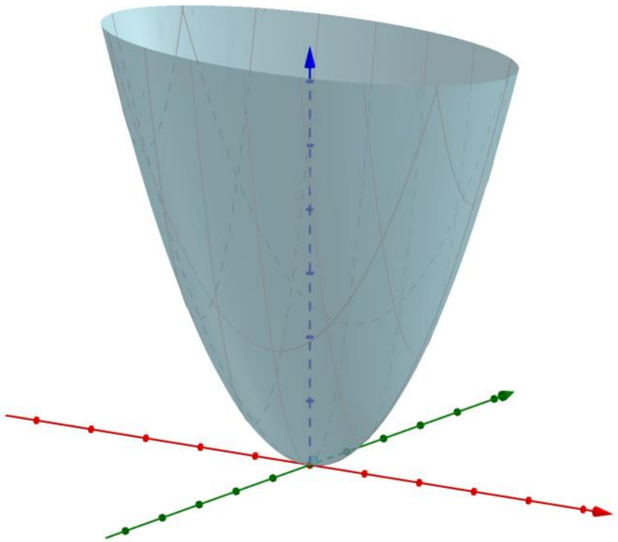


$$\phi(x) = \gamma x^2 = x\gamma x$$
$$\gamma > 0$$

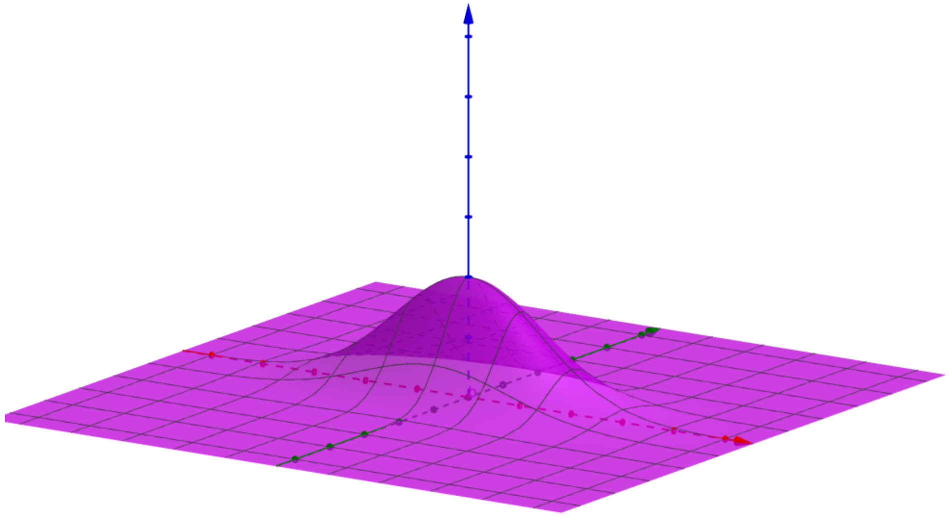


$$\rho(x) \propto e^{-\phi(x)} = e^{-x\gamma x}$$
$$\gamma = 1/\sigma^2$$

# Normal Distribution



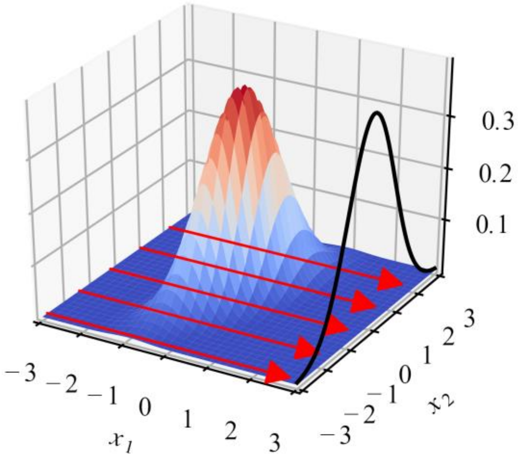
$$\phi(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\Gamma} \boldsymbol{x}$$
$$\boldsymbol{\Gamma} \succcurlyeq 0$$



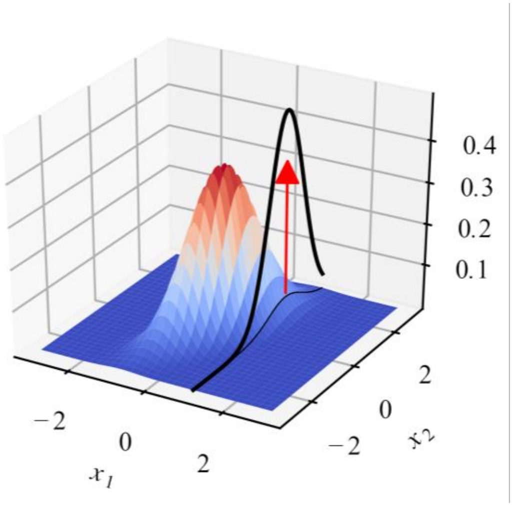
$$\rho(\boldsymbol{x}) \propto e^{-\phi(\boldsymbol{x})} = e^{-\boldsymbol{x}^\top \boldsymbol{\Gamma} \boldsymbol{x}}$$
$$\boldsymbol{\Gamma} = \boldsymbol{\Sigma}^{-1}$$

# Marginalizing and Conditioning Normals

$$\begin{pmatrix} \boldsymbol{z}_P \\ \boldsymbol{z}_Q \end{pmatrix} \sim \mathcal{N}\left( \begin{bmatrix} \bar{\boldsymbol{z}}_P \\ \bar{\boldsymbol{z}}_Q \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{\mathrm{PP}} \boldsymbol{\Sigma}_{\mathrm{PQ}} \\ \boldsymbol{\Sigma}_{\mathrm{QP}} \boldsymbol{\Sigma}_{\mathrm{QQ}} \end{bmatrix} \right)$$



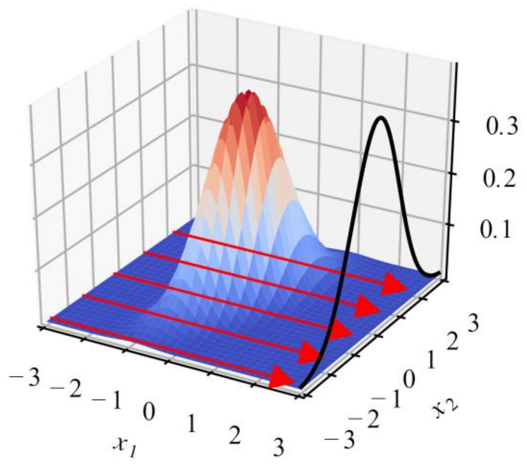

Marginalizing




Conditioning

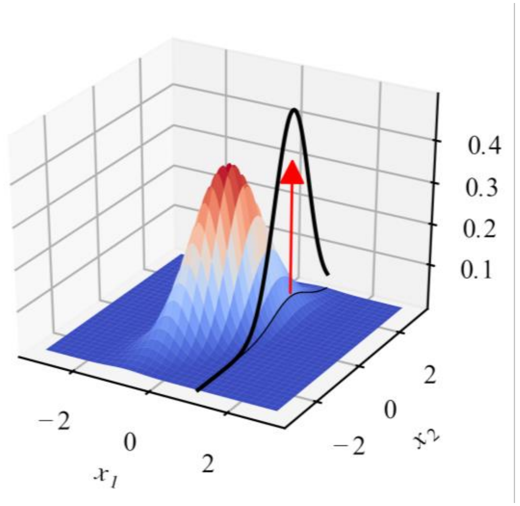
# Marginalizing and Conditioning Normals

$$\begin{pmatrix} \boldsymbol{z}_P \\ \boldsymbol{z}_Q \end{pmatrix} \sim \mathcal{N}\left( \begin{bmatrix} \bar{\boldsymbol{z}}_P \\ \bar{\boldsymbol{z}}_Q \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{\mathrm{PP}} \boldsymbol{\Sigma}_{\mathrm{PQ}} \\ \boldsymbol{\Sigma}_{\mathrm{QP}} \boldsymbol{\Sigma}_{\mathrm{QQ}} \end{bmatrix} \right)$$




Marginalizing

$\boldsymbol{z}_P \sim \mathcal{N}(\bar{\boldsymbol{z}}_P, \boldsymbol{\Sigma}_{PP})$

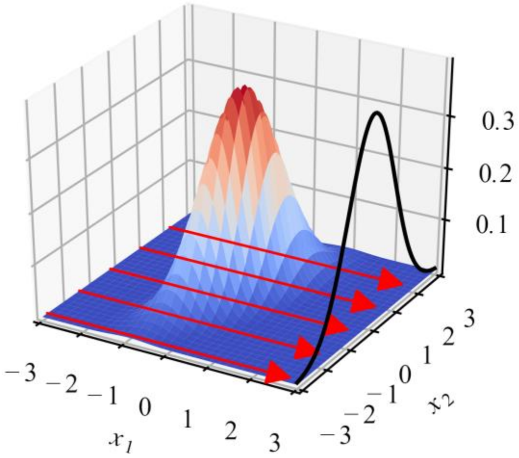



Conditioning

$\boldsymbol{z}_P | \boldsymbol{z}_Q \sim \mathcal{N}(\bar{\boldsymbol{z}}_P + \boldsymbol{\Sigma}_{PQ}\boldsymbol{\Sigma}_{QQ}^{-1}(\boldsymbol{z}_Q - \bar{\boldsymbol{z}}_Q),$
$\boldsymbol{\Sigma}_{PP} - \boldsymbol{\Sigma}_{PQ}\boldsymbol{\Sigma}_{QQ}^{-1}\boldsymbol{\Sigma}_{QP})$

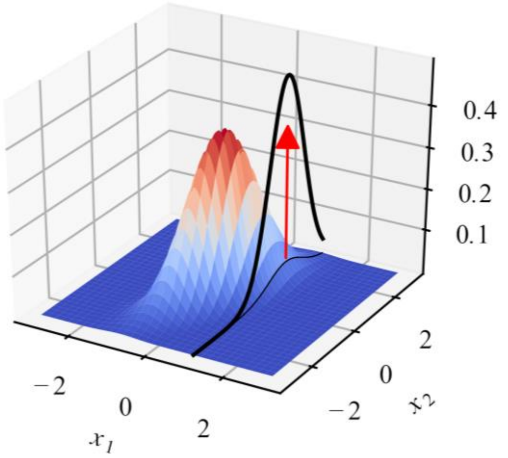
# Marginalizing and Conditioning Normals

$$\begin{pmatrix} \boldsymbol{z}_P \\ \boldsymbol{z}_Q \end{pmatrix} \sim \mathcal{N}\left( \begin{bmatrix} \bar{\boldsymbol{z}}_P \\ \bar{\boldsymbol{z}}_Q \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Gamma}_{\mathrm{PP}} \boldsymbol{\Gamma}_{\mathrm{PQ}} \\ \boldsymbol{\Gamma}_{\mathrm{QP}} \boldsymbol{\Gamma}_{\mathrm{QQ}} \end{bmatrix}^{-1} \right)$$




Marginalizing

$$\boldsymbol{z}_P \sim \mathcal{N}\left( \bar{\boldsymbol{z}}_P, \left( \boldsymbol{\Gamma}_{PP} - \boldsymbol{\Gamma}_{PQ} \boldsymbol{\Gamma}_{QQ}^{-1} \boldsymbol{\Gamma}_{QP} \right)^{-1} \right)$$




Conditioning

$$\boldsymbol{z}_P | \boldsymbol{z}_Q \sim \mathcal{N}\left( \bar{\boldsymbol{z}}_P - \boldsymbol{\Gamma}_{PP}^{-1} \boldsymbol{\Gamma}_{PQ} (\boldsymbol{z}_Q - \bar{\boldsymbol{z}}_Q), \boldsymbol{\Gamma}_{PP}^{-1} \right)$$

# Unifying Model

Distribution over attribute matrix:

$$\rho(\boldsymbol{A} = \boldsymbol{A} | \boldsymbol{H}, \boldsymbol{h}) \propto e^{-\phi(\boldsymbol{A} | \boldsymbol{H}, \boldsymbol{h})}$$

- $\boldsymbol{H} \in \mathbb{R}^{(p+1)\times(p+1)}$ is positive definite
- $\boldsymbol{h} \in \mathbb{R}^{(p+1)}$ is entrywise positive

$$\phi(\boldsymbol{A} | \boldsymbol{H}, \boldsymbol{h}) = \frac{1}{2} \sum_{u=1}^{n} \boldsymbol{a}_u^\top \boldsymbol{H} \boldsymbol{a}_u + \frac{1}{2} \sum_{i=1}^{p+1} h_i \boldsymbol{A}_i^\top \boldsymbol{N} \boldsymbol{A}_i$$

# Unifying Model

Distribution over attribute matrix:

$$\rho(\boldsymbol{A} = \boldsymbol{A} | \boldsymbol{H}, \boldsymbol{h}) \propto e^{-\phi(\boldsymbol{A}|\boldsymbol{H},\boldsymbol{h})}$$

- $\boldsymbol{H} \in \mathbb{R}^{(p+1)\times(p+1)}$ is positive definite
- $\boldsymbol{h} \in \mathbb{R}^{(p+1)}$ is entrywise positive

$$\phi(\boldsymbol{A}|\boldsymbol{H},\boldsymbol{h}) = \underbrace{\frac{1}{2}\sum_{u=1}^{n} \boldsymbol{a}_u^\top \boldsymbol{H} \boldsymbol{a}_u}_{\text{Correlations among attributes in each node}} + \frac{1}{2}\sum_{i=1}^{p+1} h_i \boldsymbol{A}_i^\top \boldsymbol{N} \boldsymbol{A}_i$$

# Unifying Model

Distribution over attribute matrix:

$$\rho(\boldsymbol{A} = \boldsymbol{A} | \boldsymbol{H}, \boldsymbol{h}) \propto e^{-\phi(\boldsymbol{A}|\boldsymbol{H},\boldsymbol{h})}$$

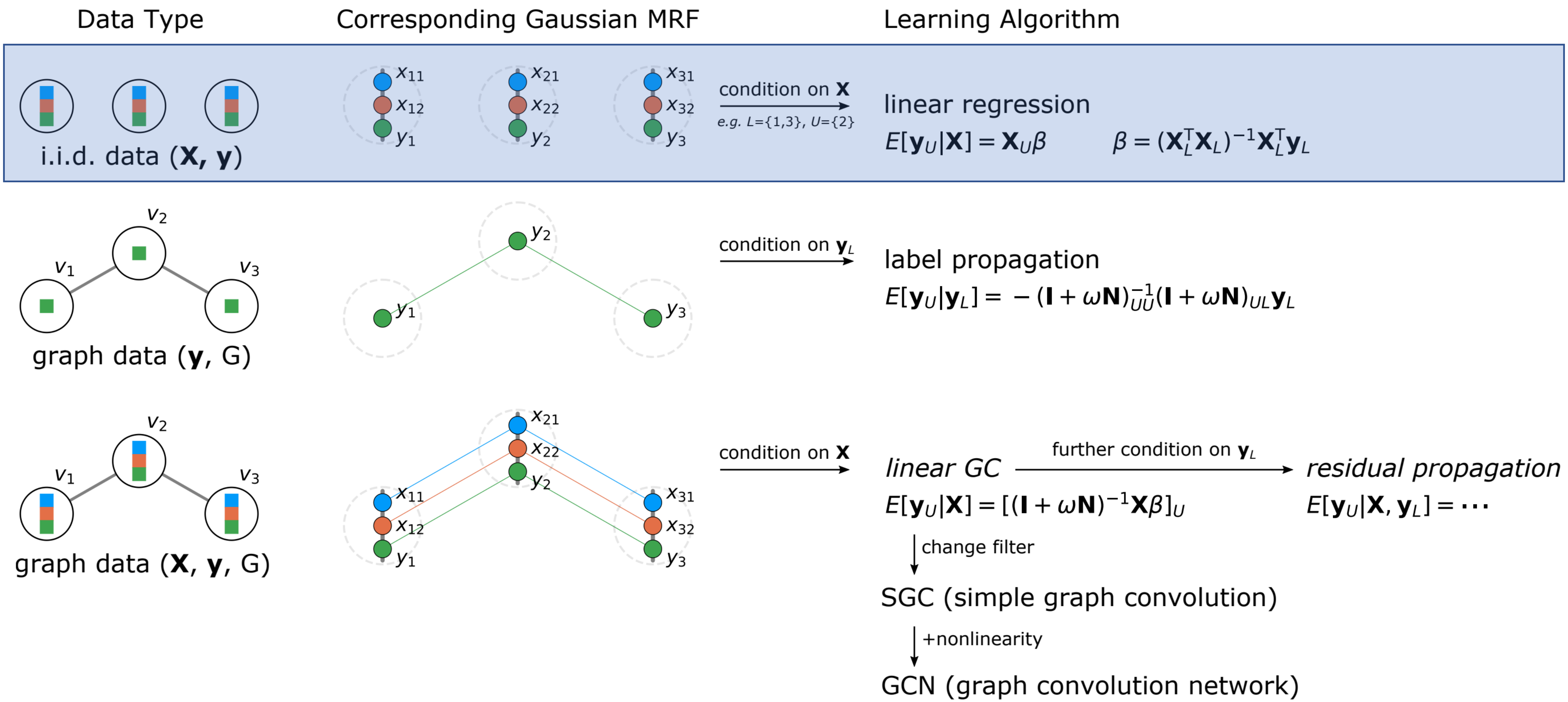
- $\boldsymbol{H} \in \mathbb{R}^{(p+1)\times(p+1)}$ is positive definite
- $\boldsymbol{h} \in \mathbb{R}^{(p+1)}$ is entrywise positive

$$\phi(\boldsymbol{A}|\boldsymbol{H},\boldsymbol{h}) = \underbrace{\frac{1}{2}\sum_{u=1}^{n} \boldsymbol{a}_u^\top \boldsymbol{H} \boldsymbol{a}_u}_{\text{Correlations among attributes in each node}} + \underbrace{\frac{1}{2}\sum_{i=1}^{p+1} h_i \boldsymbol{A}_i^\top \boldsymbol{N} \boldsymbol{A}_i}_{\text{Discourage roughness of each attribute over graph}}$$

Correlations among attributes in each node

Discourage roughness of each attribute over graph

$$\boldsymbol{A}_i^\top \boldsymbol{N} \boldsymbol{A}_i = \sum_{(u,v)\in E} \left( \frac{A_{ui}}{\sqrt{d_u}} - \frac{A_{vi}}{\sqrt{d_v}} \right)^2$$



Data Type
Corresponding Gaussian MRF
Learning Algorithm
i.i.d. data (X, y)
$x_{11}$ $x_{12}$ $y_1$ $x_{21}$ $x_{22}$ $y_2$ $x_{31}$ $x_{32}$ $y_3$
condition on X
e.g. L={1,3}, U={2}
linear regression
$E[\mathbf{y}_U|\mathbf{X}] = \mathbf{X}_U\beta$ $\beta = (\mathbf{X}_L^T\mathbf{X}_L)^{-1}\mathbf{X}_L^T\mathbf{y}_L$
$v_1$ $v_2$ $v_3$
graph data (y, G)
condition on $\mathbf{y}_L$
label propagation
$E[\mathbf{y}_U|\mathbf{y}_L] = -(\mathbf{I}+\omega\mathbf{N})_{UU}^{-1}(\mathbf{I}+\omega\mathbf{N})_{UL}\mathbf{y}_L$
graph data (X, y, G)
condition on X
linear GC
further condition on $\mathbf{y}_L$
residual propagation
$E[\mathbf{y}_U|\mathbf{X}] = [(\mathbf{I}+\omega\mathbf{N})^{-1}\mathbf{X}\beta]_U$
$E[\mathbf{y}_U|\mathbf{X},\mathbf{y}_L] = \cdots$
change filter
SGC (simple graph convolution)
+nonlinearity
GCN (graph convolution network)

# Edgeless Case

$$\phi(\boldsymbol{A}|\boldsymbol{H},\boldsymbol{h}) = \frac{1}{2}\sum_{u=1}^{n} \boldsymbol{a}_u^{\top}\boldsymbol{H}\boldsymbol{a}_u + \frac{1}{2}\sum_{i=1}^{p+1} h_i \boldsymbol{A}_i^{\top}\boldsymbol{N}\boldsymbol{A}_i$$

# Edgeless Case

$$\phi(\boldsymbol{A}|\boldsymbol{H},\boldsymbol{h}) = \frac{1}{2}\sum_{u=1}^{n} \boldsymbol{a}_u^\top \boldsymbol{H}\boldsymbol{a}_u + \frac{1}{2}\sum_{i=1}^{p+1} h_i \boldsymbol{A}_i^\top \boldsymbol{N}\boldsymbol{A}_i$$

$$\rho(\boldsymbol{A} = \boldsymbol{A}|\boldsymbol{H},\boldsymbol{h}) = \frac{e^{-\frac{1}{2}\sum_{u=1}^{n} \boldsymbol{a}_u^\top \boldsymbol{H}\boldsymbol{a}_u}}{\int d\boldsymbol{A}' e^{-\frac{1}{2}\sum_{u=1}^{n} \boldsymbol{a'}_u^\top \boldsymbol{H}\boldsymbol{a}'_u}}$$

$$= \prod_{u=1}^{n} \frac{e^{-\frac{1}{2}\boldsymbol{a}_u^\top \boldsymbol{H}\boldsymbol{a}_u}}{\int d\boldsymbol{a}' e^{-\frac{1}{2}\boldsymbol{a'}_u^\top \boldsymbol{H}\boldsymbol{a}'_u}}$$

$$\propto \prod_{u=1}^{n} e^{-\frac{1}{2}\boldsymbol{a}_u^\top \boldsymbol{H}\boldsymbol{a}_u}.$$

# Edgeless Case

$$\phi(\boldsymbol{A}|\boldsymbol{H},\boldsymbol{h}) = \frac{1}{2}\sum_{u=1}^{n} \boldsymbol{a}_u^\top \boldsymbol{H} \boldsymbol{a}_u + \frac{1}{2}\sum_{i=1}^{p+1} h_i \boldsymbol{A}_i^\top \boldsymbol{N} \boldsymbol{A}_i$$

$$\rho(\boldsymbol{A} = \boldsymbol{A}|\boldsymbol{H},\boldsymbol{h}) = \frac{e^{-\frac{1}{2}\sum_{u=1}^{n} \boldsymbol{a}_u^\top \boldsymbol{H} \boldsymbol{a}_u}}{\int d\boldsymbol{A}' e^{-\frac{1}{2}\sum_{u=1}^{n} \boldsymbol{a}'^\top_u \boldsymbol{H} \boldsymbol{a}'_u}}$$

$$= \prod_{u=1}^{n} \frac{e^{-\frac{1}{2}\boldsymbol{a}_u^\top \boldsymbol{H} \boldsymbol{a}_u}}{\int d\boldsymbol{a}' e^{-\frac{1}{2}\boldsymbol{a}'^\top_u \boldsymbol{H} \boldsymbol{a}'_u}}$$

$$\propto \prod_{u=1}^{n} e^{-\frac{1}{2}\boldsymbol{a}_u^\top \boldsymbol{H} \boldsymbol{a}_u}.$$

$\rho$ decomposes into a product of $n$ IID normals, one for each node, with mean $0$ and precision $H$.

# Edgeless Case: Conditioning

The rows $\{\boldsymbol{a}_u\}_{u=1}^{n}$ of the attribute matrix are IID normals with mean $0$, precision $H$.

Now condition on the features $\boldsymbol{X}$ to find the expectation of the labels $\boldsymbol{y}$:

For each node $u \in V$, $\mathbb{E}[y_u | \boldsymbol{X} = \boldsymbol{X}] = \underbrace{\mathbb{E}[y_u | \boldsymbol{x}_u = \boldsymbol{x}_u]}_{\text{Condition on } p \text{ out of } p+1 \text{ attributes}}$

# Edgeless Case: Conditioning

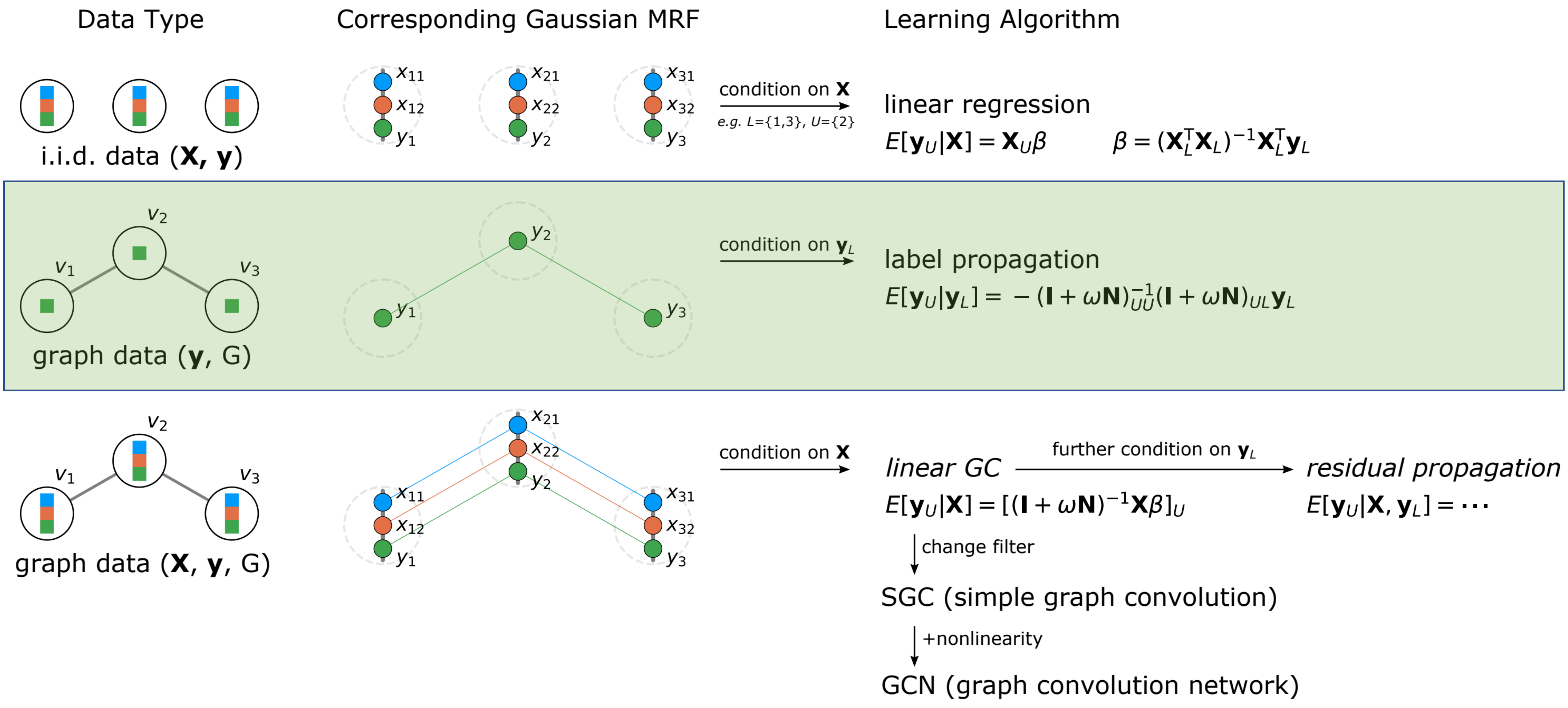
The rows $\{\boldsymbol{a}_u\}_{u=1}^{n}$ of the attribute matrix are IID normals with mean $0$, precision $H$.

Now condition on the features $\boldsymbol{X}$ to find the expectation of the labels $\boldsymbol{y}$:

For each node $u \in V$, $\mathbb{E}[y_u | \boldsymbol{X} = \boldsymbol{X}] = \mathbb{E}[y_u | \boldsymbol{x}_u = \boldsymbol{x}_u] = \boldsymbol{x}_u^\top \left( \frac{-\boldsymbol{H}_{1:p,p+1}}{H_{p+1,p+1}} \right)$.

$\mathbb{E}[\boldsymbol{y} | \boldsymbol{X} = \boldsymbol{X}] = \boldsymbol{X}\boldsymbol{\beta}$ for $\boldsymbol{\beta} = \frac{-\boldsymbol{H}_{1:p,p+1}}{H_{p+1,p+1}} \in \mathbb{R}^p$.

Algorithm: Rather than finding $\boldsymbol{H}$, fit $\boldsymbol{\beta}$ directly via linear regression on known labels $(\boldsymbol{X}_L, \boldsymbol{y}_L)$.


Data Type
Corresponding Gaussian MRF
Learning Algorithm
i.i.d. data (X, y)
condition on X
e.g. L={1,3}, U={2}
linear regression
$E[\mathbf{y}_U|\mathbf{X}] = \mathbf{X}_U\beta$
$\beta = (\mathbf{X}_L^T\mathbf{X}_L)^{-1}\mathbf{X}_L^T\mathbf{y}_L$
graph data (y, G)
condition on $\mathbf{y}_L$
label propagation
$E[\mathbf{y}_U|\mathbf{y}_L] = -(\mathbf{I}+\omega\mathbf{N})_{UU}^{-1}(\mathbf{I}+\omega\mathbf{N})_{UL}\mathbf{y}_L$
graph data (X, y, G)
condition on X
linear GC
further condition on $\mathbf{y}_L$
residual propagation
$E[\mathbf{y}_U|\mathbf{X}] = [(\mathbf{I}+\omega\mathbf{N})^{-1}\mathbf{X}\beta]_U$
$E[\mathbf{y}_U|\mathbf{X},\mathbf{y}_L] = \cdots$
change filter
SGC (simple graph convolution)
+nonlinearity
GCN (graph convolution network)

# Featureless Case

General model:

$$\phi(\boldsymbol{A}|\boldsymbol{H},\boldsymbol{h}) = \frac{1}{2}\sum_{u=1}^{n} \boldsymbol{a}_u^{\top}\boldsymbol{H}\boldsymbol{a}_u + \frac{1}{2}\sum_{i=1}^{p+1} h_i \boldsymbol{A}_i^{\top}\boldsymbol{N}\boldsymbol{A}_i$$

Without features ($p = 0$):

$$\phi(\boldsymbol{y}|H,h) = \frac{1}{2}\sum_{u=1}^{n} y_u H y_u + \frac{1}{2} h \boldsymbol{y}^{\top}\boldsymbol{N}\boldsymbol{y}$$

# Featureless Case

General model:

$$\phi(\boldsymbol{A}|\boldsymbol{H},\boldsymbol{h}) = \frac{1}{2}\sum_{u=1}^{n}\boldsymbol{a}_u^\top\boldsymbol{H}\boldsymbol{a}_u + \frac{1}{2}\sum_{i=1}^{p+1}h_i\boldsymbol{A}_i^\top\boldsymbol{N}\boldsymbol{A}_i$$

Without features ($p = 0$):

$$\begin{aligned}\phi(\boldsymbol{y}|H,h) &= \frac{1}{2}\sum_{u=1}^{n}y_u H y_u + \frac{1}{2}h\boldsymbol{y}^\top\boldsymbol{N}\boldsymbol{y}\\ &= \frac{1}{2}H\boldsymbol{y}^\top\boldsymbol{I}_n\boldsymbol{y} + \frac{1}{2}h\boldsymbol{y}^\top\boldsymbol{N}\boldsymbol{y}\\ &= \tfrac{1}{2}\boldsymbol{y}^\top(H\boldsymbol{I}_n + h\boldsymbol{N})\boldsymbol{y} = \tfrac{1}{2}\boldsymbol{y}^\top\boldsymbol{\Gamma}\boldsymbol{y}, \text{ where } \boldsymbol{\Gamma} = H\boldsymbol{I}_n + h\boldsymbol{N}.\end{aligned}$$

# Featureless Case: Conditioning

$\rho(\boldsymbol{y} = \boldsymbol{y} | H, h) = e^{-\frac{1}{2}\boldsymbol{y}^\top \boldsymbol{\Gamma} \boldsymbol{y}}$, where $\boldsymbol{\Gamma} = H\boldsymbol{I}_n + h\boldsymbol{N}$.

Now condition on known labels $\boldsymbol{y}_L$ to find unknown labels $\boldsymbol{y}_U$:

# Featureless Case: Conditioning

$\rho(\boldsymbol{y} = \boldsymbol{y} | H, h) = e^{-\frac{1}{2}\boldsymbol{y}^\top \boldsymbol{\Gamma} \boldsymbol{y}}$, where $\boldsymbol{\Gamma} = H\boldsymbol{I}_n + h\boldsymbol{N}$.

Now condition on known labels $\boldsymbol{y}_L$ to find unknown labels $\boldsymbol{y}_U$:

$$\begin{aligned}
\mathbb{E}[\boldsymbol{y}_U | \boldsymbol{y}_L = \boldsymbol{y}_L] &= -\boldsymbol{\Gamma}_{UU}^{-1} \boldsymbol{\Gamma}_{UL} \boldsymbol{y}_L \\
&= -(H\boldsymbol{I}_n + h\boldsymbol{N})_{UU}^{-1} (H\boldsymbol{I}_n + h\boldsymbol{N})_{UL} \boldsymbol{y}_L \\
&= -(\boldsymbol{I}_n + \omega\boldsymbol{N})_{UU}^{-1} (\boldsymbol{I}_n + \omega\boldsymbol{N})_{UL} \boldsymbol{y}_L, \text{ where } \omega = h/H.
\end{aligned}$$

# Featureless Case: Conditioning

$\rho(\boldsymbol{y} = \boldsymbol{y} | H, h) = e^{-\frac{1}{2}\boldsymbol{y}^{\top}\boldsymbol{\Gamma}\boldsymbol{y}}$, where $\boldsymbol{\Gamma} = H\boldsymbol{I}_n + h\boldsymbol{N}$.

Now condition on known labels $\boldsymbol{y}_L$ to find unknown labels $\boldsymbol{y}_U$:

$\mathbb{E}[\boldsymbol{y}_U | \boldsymbol{y}_L = \boldsymbol{y}_L] = -(\boldsymbol{I}_n + \omega\boldsymbol{N})_{UU}^{-1}(\boldsymbol{I}_n + \omega\boldsymbol{N})_{UL}\boldsymbol{y}_L$, where $\omega = h/H$.

# Featureless Case: Conditioning

$\rho(\boldsymbol{y} = \boldsymbol{y}|H, h) = e^{-\frac{1}{2}\boldsymbol{y}^\top \boldsymbol{\Gamma} \boldsymbol{y}}$, where $\boldsymbol{\Gamma} = H\boldsymbol{I}_n + h\boldsymbol{N}$.

Now condition on known labels $\boldsymbol{y}_L$ to find unknown labels $\boldsymbol{y}_U$:

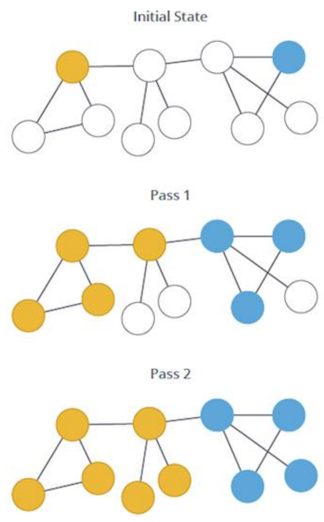
$\mathbb{E}[\boldsymbol{y}_U|\boldsymbol{y}_L = \boldsymbol{y}_L] = -(\boldsymbol{I}_n + \omega\boldsymbol{N})^{-1}_{UU}(\boldsymbol{I}_n + \omega\boldsymbol{N})_{UL}\boldsymbol{y}_L$, where $\omega = h/H$.

Algorithm: Label propagation.

$\forall u \in L, y_u^{(0)} = y_u$ and $y_u^{(t+1)} = y_u^{(t)}$,

$\forall u \in U, y_u^{(0)} = 0$ and $y_u^{(t+1)} = (1-\alpha)y_u^{(0)} + \alpha d_u^{-1/2} \sum_{v \in N_1(u)} d_v^{-1/2} y_v^{(t)}$,

where $\alpha = \frac{\omega}{1+\omega} \in (0,1)$ and $N_1(u)$ are the neighbors of $u$.



# Featureless Case: Conditioning

$\rho(\boldsymbol{y} = \boldsymbol{y}|H, h) = e^{-\frac{1}{2}\boldsymbol{y}^\top \boldsymbol{\Gamma} \boldsymbol{y}}$, where $\boldsymbol{\Gamma} = H\boldsymbol{I}_n + h\boldsymbol{N}$.

Now condition on known labels $\boldsymbol{y}_L$ to find unknown labels $\boldsymbol{y}_U$:

$\mathbb{E}[\boldsymbol{y}_U|\boldsymbol{y}_L = \boldsymbol{y}_L] = -(\boldsymbol{I}_n + \omega\boldsymbol{N})^{-1}_{UU}(\boldsymbol{I}_n + \omega\boldsymbol{N})_{UL}\boldsymbol{y}_L$, where $\omega = h/H$.

Algorithm: Label propagation.

$\forall u \in L, y_u^{(0)} = y_u$ and $y_u^{(t+1)} = y_u^{(t)}$,

$\forall u \in U, y_u^{(0)} = 0$ and $y_u^{(t+1)} = (1-\alpha)y_u^{(0)} + \alpha d_u^{-1/2} \sum_{v \in N_1(u)} d_v^{-1/2} y_v^{(t)}$,

where $\alpha = \frac{\omega}{1+\omega} \in (0,1)$ and $N_1(u)$ are the neighbors of $u$.

Equivalent?

# Featureless Case: Label Prop Proof

Vectorized form of iteration

$\forall u \in L, y_u^{(0)} = y_u$ and $y_u^{(t+1)} = y_u^{(t)}$

$\forall u \in U, y_u^{(0)} = 0$ and $y_u^{(t+1)} = (1-\alpha) y_u^{(0)} + \alpha d_u^{-1/2} \sum_{v \in N_1(u)} d_v^{-1/2} y_v^{(t)}$.

$$\boldsymbol{y}_U^{(t+1)} = (1-\alpha)\boldsymbol{y}_U^{(0)} + \alpha \boldsymbol{S}_{U,U\cup L} \boldsymbol{y}^{(t)}$$

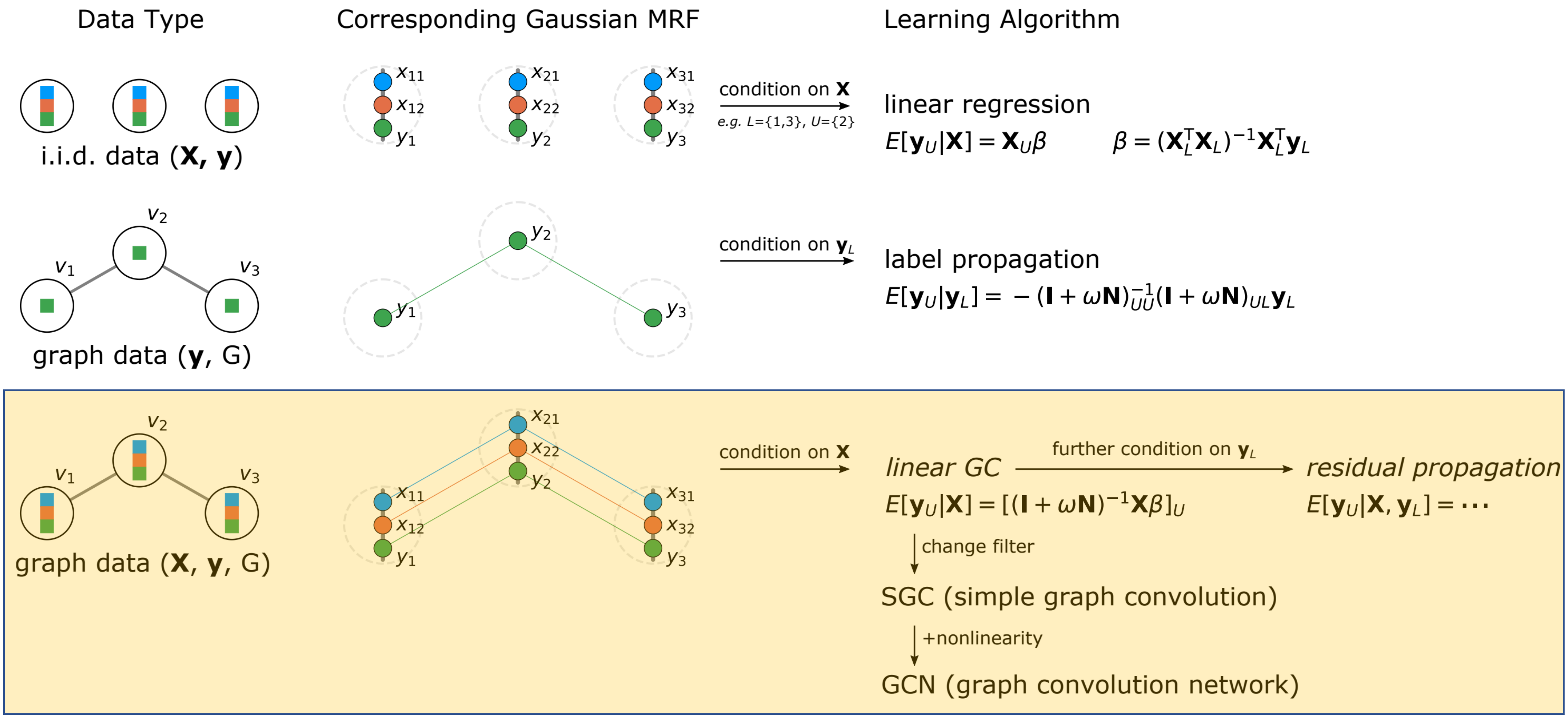
$$= (1-\alpha)\boldsymbol{y}_U^{(0)} + \alpha(\boldsymbol{S}_{UU}\boldsymbol{y}_U^{(t)} + \boldsymbol{S}_{UL}\boldsymbol{y}_L^{(t)})$$

$$= \alpha \boldsymbol{S}_{UU}\boldsymbol{y}_U^{(t)} + \alpha \boldsymbol{S}_{UL}\boldsymbol{y}_L.$$

# Featureless Case: Label Prop Proof

Fixed point of iteration $\boldsymbol{y}_U^{(t+1)} = \alpha \boldsymbol{S}_{UU} \boldsymbol{y}_U^{(t)} + \alpha \boldsymbol{S}_{UL} \boldsymbol{y}_L$.

$$
\begin{aligned}
\boldsymbol{y}_U^{(\infty)} &= \alpha \boldsymbol{S}_{UU} \boldsymbol{y}_U^{(\infty)} + \alpha \boldsymbol{S}_{UL} \boldsymbol{y}_L. \\
&= (\boldsymbol{I} - \alpha \boldsymbol{S})_{UU}^{-1} (\alpha \boldsymbol{S}_{UL}) \boldsymbol{y}_L \\
&= -(\boldsymbol{I} - \alpha \boldsymbol{S})_{UU}^{-1} (-\alpha \boldsymbol{S}_{UL}) \boldsymbol{y}_L \\
&= -(\boldsymbol{I} - \alpha \boldsymbol{S})_{UU}^{-1} (\boldsymbol{I} - \alpha \boldsymbol{S})_{UL} \boldsymbol{y}_L \\
&= -\big((1-\alpha)\boldsymbol{I} + \alpha \boldsymbol{I} - \alpha \boldsymbol{S}\big)_{UU}^{-1} \big((1-\alpha)\boldsymbol{I} + \alpha \boldsymbol{I} - \alpha \boldsymbol{S}\big)_{UL} \boldsymbol{y}_L \\
&= -\left(\boldsymbol{I} + \frac{\alpha}{1-\alpha}(\boldsymbol{I} - \boldsymbol{S})\right)_{UU}^{-1} \left(\boldsymbol{I} + \frac{\alpha}{1-\alpha}(\boldsymbol{I} - \boldsymbol{S})\right)_{UL} \boldsymbol{y}_L \\
&= -(\boldsymbol{I} + \omega \boldsymbol{N})_{UU}^{-1} (\boldsymbol{I} + \omega \boldsymbol{N})_{UL} \boldsymbol{y}_L.
\end{aligned}
$$


Data Type
Corresponding Gaussian MRF
Learning Algorithm
$x_{11}$ $x_{12}$ $y_1$ $x_{21}$ $x_{22}$ $y_2$ $x_{31}$ $x_{32}$ $y_3$
i.i.d. data (**X**, **y**)
condition on **X**
e.g. L={1,3}, U={2}
linear regression
$E[\mathbf{y}_U|\mathbf{X}] = \mathbf{X}_U\beta \qquad \beta = (\mathbf{X}_L^T\mathbf{X}_L)^{-1}\mathbf{X}_L^T\mathbf{y}_L$
$v_1$ $v_2$ $v_3$
graph data (**y**, G)
condition on $\mathbf{y}_L$
label propagation
$E[\mathbf{y}_U|\mathbf{y}_L] = -(\mathbf{I}+\omega\mathbf{N})_{UU}^{-1}(\mathbf{I}+\omega\mathbf{N})_{UL}\mathbf{y}_L$
graph data (**X**, **y**, G)
condition on **X**
*linear GC*
further condition on $\mathbf{y}_L$
*residual propagation*
$E[\mathbf{y}_U|\mathbf{X}] = [(\mathbf{I}+\omega\mathbf{N})^{-1}\mathbf{X}\beta]_U$
$E[\mathbf{y}_U|\mathbf{X},\mathbf{y}_L] = \cdots$
change filter
SGC (simple graph convolution)
+nonlinearity
GCN (graph convolution network)

# Graph + Features Case

$$\phi(\boldsymbol{A}|\boldsymbol{H},\boldsymbol{h}) = \frac{1}{2}\sum_{u=1}^{n}\boldsymbol{a}_u^\top \boldsymbol{H}\boldsymbol{a}_u + \frac{1}{2}\sum_{i=1}^{p+1} h_i \boldsymbol{A}_i^\top \boldsymbol{N}\boldsymbol{A}_i$$

$\rho(\boldsymbol{A} = \boldsymbol{A}|\boldsymbol{H},\boldsymbol{h}) = e^{-\frac{1}{2}\left(\mathrm{vec}(\boldsymbol{A})\right)^\top \boldsymbol{\Gamma}\left(\mathrm{vec}(\boldsymbol{A})\right)}$, where $\boldsymbol{\Gamma} = \boldsymbol{H} \otimes \boldsymbol{I}_n + \mathrm{diag}(\boldsymbol{h}) \otimes \boldsymbol{N}$.

Now condition on features $\mathbf{X}$ to find labels $\boldsymbol{y}$:

# Graph + Features Case

$$\phi(\boldsymbol{A}|\boldsymbol{H},\boldsymbol{h}) = \frac{1}{2}\sum_{u=1}^{n} \boldsymbol{a}_u^\top \boldsymbol{H}\boldsymbol{a}_u + \frac{1}{2}\sum_{i=1}^{p+1} h_i \boldsymbol{A}_i^\top \boldsymbol{N}\boldsymbol{A}_i$$

$\rho(\boldsymbol{A}=\boldsymbol{A}|\boldsymbol{H},\boldsymbol{h}) = e^{-\frac{1}{2}(\mathrm{vec}(\boldsymbol{A}))^\top \boldsymbol{\Gamma}(\mathrm{vec}(\boldsymbol{A}))}$, where $\boldsymbol{\Gamma} = \boldsymbol{H}\otimes\boldsymbol{I}_n + \mathrm{diag}(\boldsymbol{h})\otimes\boldsymbol{N}$.

Now condition on features $\mathbf{X}$ to find labels $\boldsymbol{y}$:

$$\begin{aligned}\mathbb{E}[\boldsymbol{y}|\boldsymbol{X}=\boldsymbol{X}] &= -\boldsymbol{\Gamma}_{UU}^{-1}\boldsymbol{\Gamma}_{UL}\mathrm{vec}(\boldsymbol{X}) \\ &= \left(H_{p+1,p+1}\boldsymbol{I}_n + h_{p+1}\boldsymbol{N}\right)^{-1}\left(-\boldsymbol{H}_{1:p,p+1}^\top \otimes \boldsymbol{I}_n\right)\mathrm{vec}(\boldsymbol{X}) \\ &= (\boldsymbol{I}_n + \omega\boldsymbol{N})^{-1}\boldsymbol{X}\boldsymbol{\beta}, \text{ where } \omega = \frac{h_{p+1}}{H_{p+1,p+1}} \text{ and } \boldsymbol{\beta} = \frac{-\boldsymbol{H}_{1:p,p+1}}{H_{p+1,p+1}}.\end{aligned}$$

# Graph + Features Case

$$\phi(\boldsymbol{A}|\boldsymbol{H},\boldsymbol{h}) = \frac{1}{2}\sum_{u=1}^{n} \boldsymbol{a}_u^\top \boldsymbol{H} \boldsymbol{a}_u + \frac{1}{2}\sum_{i=1}^{p+1} h_i \boldsymbol{A}_i^\top \boldsymbol{N} \boldsymbol{A}_i$$

$\rho(\boldsymbol{A} = \boldsymbol{A}|\boldsymbol{H},\boldsymbol{h}) = e^{-\frac{1}{2}(\mathrm{vec}(\boldsymbol{A}))^\top \boldsymbol{\Gamma} (\mathrm{vec}(\boldsymbol{A}))}$, where $\boldsymbol{\Gamma} = \boldsymbol{H} \otimes \boldsymbol{I}_n + \mathrm{diag}(\boldsymbol{h}) \otimes \boldsymbol{N}$.

Now condition on features $\mathbf{X}$ to find labels $\boldsymbol{y}$:

$\mathbb{E}[\boldsymbol{y}|\boldsymbol{X} = \boldsymbol{X}] = (\boldsymbol{I}_n + \omega \boldsymbol{N})^{-1} \boldsymbol{X} \boldsymbol{\beta}$, where $\omega = \frac{h_{p+1}}{H_{p+1,p+1}}$ and $\boldsymbol{\beta} = \frac{-\boldsymbol{H}_{1:p,p+1}}{H_{p+1,p+1}}$.

# Graph + Features Case

$$\phi(\boldsymbol{A}|\boldsymbol{H},\boldsymbol{h}) = \frac{1}{2}\sum_{u=1}^{n} \boldsymbol{a}_u^\top \boldsymbol{H} \boldsymbol{a}_u + \frac{1}{2}\sum_{i=1}^{p+1} h_i \boldsymbol{A}_i^\top \boldsymbol{N} \boldsymbol{A}_i$$

$\rho(\boldsymbol{A} = \boldsymbol{A}|\boldsymbol{H},\boldsymbol{h}) = e^{-\frac{1}{2}(\mathrm{vec}(\boldsymbol{A}))^\top \boldsymbol{\Gamma}(\mathrm{vec}(\boldsymbol{A}))}$, where $\boldsymbol{\Gamma} = \boldsymbol{H} \otimes \boldsymbol{I}_n + \mathrm{diag}(\boldsymbol{h}) \otimes \boldsymbol{N}$.

Now condition on features $\mathbf{X}$ to find labels $\boldsymbol{y}$:

$\mathbb{E}[\boldsymbol{y}|\boldsymbol{X} = \boldsymbol{X}] = \underbrace{(\boldsymbol{I}_n + \omega\boldsymbol{N})^{-1}\boldsymbol{X}}\boldsymbol{\beta}$, where $\omega = \frac{h_{p+1}}{H_{p+1,p+1}}$ and $\boldsymbol{\beta} = \frac{-\boldsymbol{H}_{1:p,p+1}}{H_{p+1,p+1}}$.

Feature Propagation:

$\forall u \in V, \boldsymbol{x}_u^{(0)} = \boldsymbol{x}_u$ and $\boldsymbol{x}_u^{(t+1)} = (1-\alpha)\boldsymbol{x}_u^{(0)} + \alpha d_u^{-1/2} \sum_{v \in N_1(u)} d_v^{-1/2} \boldsymbol{x}_v^{(t)}$.

# Graph + Features Case

$$\phi(\boldsymbol{A}|\boldsymbol{H},\boldsymbol{h}) = \frac{1}{2}\sum_{u=1}^{n} \boldsymbol{a}_u^\top \boldsymbol{H} \boldsymbol{a}_u + \frac{1}{2}\sum_{i=1}^{p+1} h_i \boldsymbol{A}_i^\top \boldsymbol{N} \boldsymbol{A}_i$$

$\rho(\boldsymbol{A} = \boldsymbol{A}|\boldsymbol{H},\boldsymbol{h}) = e^{-\frac{1}{2}(\mathrm{vec}(\boldsymbol{A}))^\top \boldsymbol{\Gamma}(\mathrm{vec}(\boldsymbol{A}))}$, where $\boldsymbol{\Gamma} = \boldsymbol{H} \otimes \boldsymbol{I}_n + \mathrm{diag}(\boldsymbol{h}) \otimes \boldsymbol{N}$.

Now condition on features $\mathbf{X}$ to find labels $\boldsymbol{y}$:

$\mathbb{E}[\boldsymbol{y}|\boldsymbol{X} = \boldsymbol{X}] = (\boldsymbol{I}_n + \omega \boldsymbol{N})^{-1} \boldsymbol{X} \boldsymbol{\beta}$, where $\omega = \frac{h_{p+1}}{H_{p+1,p+1}}$ and $\boldsymbol{\beta} = \frac{-\boldsymbol{H}_{1:p,p+1}}{H_{p+1,p+1}}$.

Algorithm (Linear Graph Convolution / LGC):

1) Do feature propagation to compute $\overline{\boldsymbol{X}} = (\boldsymbol{I}_n + \omega \boldsymbol{N})^{-1} \boldsymbol{X}$.

2) Fit $\boldsymbol{\beta}$ directly via linear regression on $(\overline{\boldsymbol{X}}_L, \boldsymbol{y}_L)$.

# Graph + Features + Known Labels Case

Condition on both features $\boldsymbol{X}$ and known labels $\boldsymbol{y}_L$ to find unknown labels $\boldsymbol{y}_U$.
$\mathbb{E}[\boldsymbol{y}_U | \boldsymbol{X} = \boldsymbol{X}, \boldsymbol{y}_L = \boldsymbol{y}_L] = \cdots$

Algorithm (LGC with Residual Propagation):

# Graph + Features + Known Labels Case

Condition on both features $\boldsymbol{X}$ and known labels $\boldsymbol{y}_L$ to find unknown labels $\boldsymbol{y}_U$.
$\mathbb{E}[\boldsymbol{y}_U | \boldsymbol{X} = \boldsymbol{X}, \boldsymbol{y}_L = \boldsymbol{y}_L] = \cdots$

Algorithm (LGC with Residual Propagation):
1) Do feature propagation to compute $\overline{\boldsymbol{X}} = (\boldsymbol{I}_n + \omega \boldsymbol{N})^{-1} \boldsymbol{X}$.
2) Fit $\boldsymbol{\beta}$ directly via linear regression on $(\overline{\boldsymbol{X}}_L, \boldsymbol{y}_L)$.

# Graph + Features + Known Labels Case

Condition on both features $\boldsymbol{X}$ and known labels $\boldsymbol{y}_L$ to find unknown labels $\boldsymbol{y}_U$.
$\mathbb{E}[\boldsymbol{y}_U | \boldsymbol{X} = \boldsymbol{X}, \boldsymbol{y}_L = \boldsymbol{y}_L] = \cdots$

Algorithm (LGC with Residual Propagation):
1) Do feature propagation to compute $\overline{\boldsymbol{X}} = (\boldsymbol{I}_n + \omega \boldsymbol{N})^{-1} \boldsymbol{X}$.
2) Fit $\boldsymbol{\beta}$ directly via linear regression on $(\overline{\boldsymbol{X}}_L, \boldsymbol{y}_L)$.
3) Compute the regression residuals on known labels: $\overline{\boldsymbol{r}}_L = \boldsymbol{y}_L - \overline{\boldsymbol{X}}_L \boldsymbol{\beta}$.
3) Do "residual propagation" of $\overline{\boldsymbol{r}}_L$ to estimate $\overline{\boldsymbol{r}}_U$.
4) Modify the LGC predictions: return $\overline{\boldsymbol{X}}_U \boldsymbol{\beta} + \overline{\boldsymbol{r}}_U$.

# Connections to Other Graph Convolution

Linear Graph Convolution (LGC):

$$\boldsymbol{y}_{LGC} = (\boldsymbol{I}_n + \omega \boldsymbol{N})^{-1} \boldsymbol{X}\boldsymbol{\beta} = (1-\alpha)(\boldsymbol{I} + \alpha \boldsymbol{S} + \alpha^2 \boldsymbol{S}^2 + \cdots)\boldsymbol{X}\boldsymbol{\beta}.$$

Simplified Graph Convolution (SGC):

$$\boldsymbol{y}_{SGC} = \widetilde{\boldsymbol{S}}^K \boldsymbol{X}\boldsymbol{\beta}.$$

Graph Convolutional Networks (GCN):

$$\boldsymbol{y}_{GCN} = \sigma\left(\widetilde{\boldsymbol{S}} \ldots \sigma\left(\widetilde{\boldsymbol{S}} \boldsymbol{X} \boldsymbol{\Theta}^{(1)}\right) \ldots \boldsymbol{\Theta}^{(K)}\right)\boldsymbol{\beta}$$

$$= \widetilde{\boldsymbol{S}}^K \boldsymbol{X}\boldsymbol{\beta}' \text{ if } \sigma(x) = x.$$

# A Unifying Generative Model for Graph Learning Algorithms