ODE on Graph:
diffusion function as example

Shikai Fang

2022/03

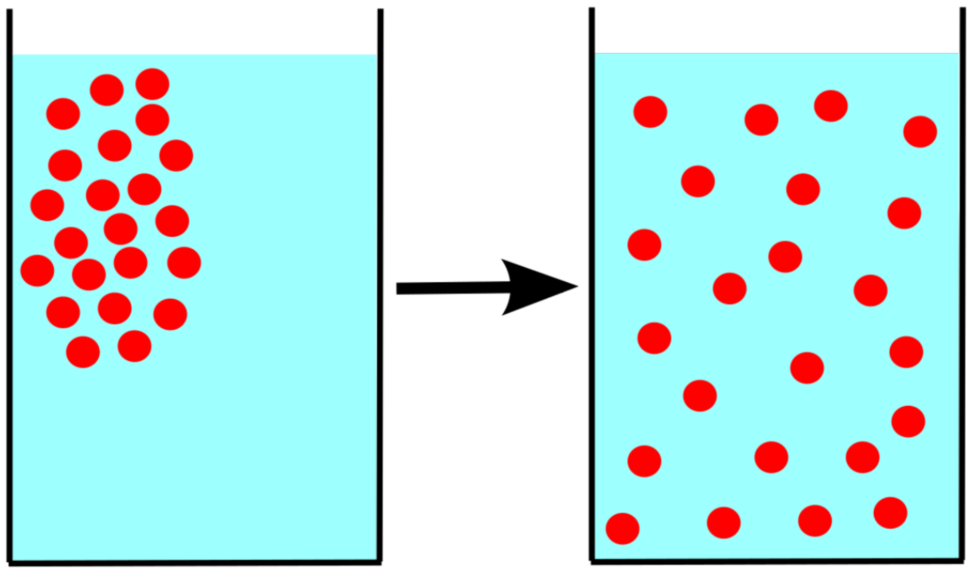
Group seminar

Material form books/paper from Simo Sarkka/Aron Solin, slides from xuecheng Li

Content

- Derivation of diffusion function
- Derivation of diffusion function on graph
- Laplacian matrix and graph spectrum
- Paper review: “Learning heat diffusion graph”
- Extended topics

Diffusion



Due to random motion, molecules of a high concentration will tend to flow towards a region in space where the concentration is lower.

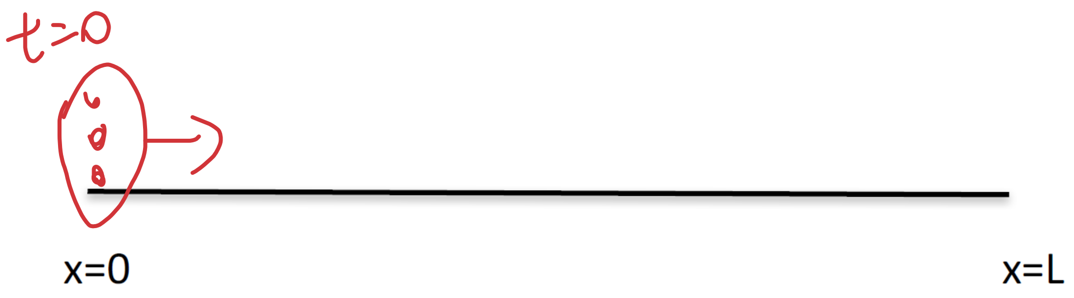
The Diffusion Equation

u: density

$u(t,x)$

1d

$\nabla\nabla$



Consider diffusion in one dimension (x) over time (t) and let u(x,t) be the concentration of the substance that is diffusing. Then

$$\boxed{\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2}}$$

Why? – derivation from scratch

is the diffusion equation with diffusion coefficient D. One would also need to supply **initial values** for u, $u(x,0)=u_0(x)$, and **boundary conditions** at each boundary.

The Diffusion Equation

To describe diffusion in a domain with more than one dimension, the second partial derivative operator is replaced with the Laplacian operator. Then the diffusion equation is,

$$\boxed{\frac{\partial u}{\partial t} = D\nabla^2 u}$$

where in three dimensions

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$$

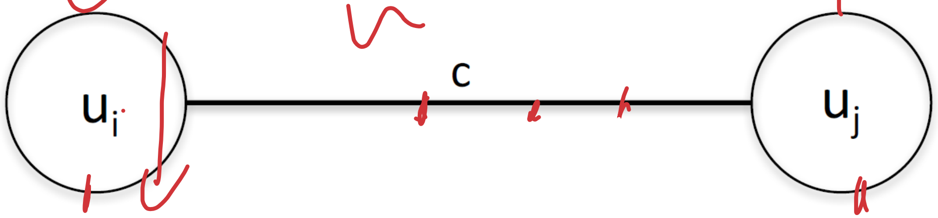
Laplacian operator

Diffusion on a Graph

What if the diffusing substance moves along edges of a graph from node to node? In this case, the domain is discrete, not a continuum.

$u_i(t)$

$t_0 =$ 5 4



$t_1 :$ 4 5

$$\frac{du}{dt} = \text{differ of flow}$$

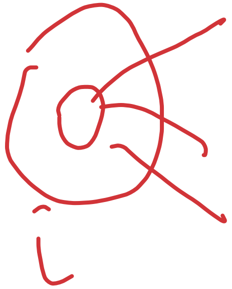
Let c be the diffusion rate across the edge, then the amount of substance that moves from node j to node i over a time period dt is c$(u_j - u_i)dt$ and from node i to node j is c$(u_i - u_j)dt$. So

$$\begin{cases} \dfrac{du_i}{dt} = c(u_j - u_i) \\ \dfrac{du_j}{dt} = c(u_i - u_j) \end{cases}$$

Diffusion on a Graph

Diffusion to and from node i must take into consideration all nodes in the graph. The connectivity of the graph is encoded in the adjacency matrix. Here we assume that we are working with a simple graph.

u_i



$$\frac{du_i}{dt} = cA_{i1}(u_1 - u_i) + cA_{i2}(u_2 - u_i) + \cdots + cA_{in}(u_n - u_i)$$

or

$$\boxed{\frac{du_i}{dt} = c\sum_{j=1}^{n} A_{ij}(u_j - u_i)}$$

$$A_{ij} = \begin{cases} 0 & \text{no edge} \\ 1 & \text{edge } ij \end{cases}$$

Diffusion on a Graph

Rewriting the last expression,

$$\frac{du_i}{dt} = c\sum_{j=1}^{n} A_{ij}u_j - cu_i \underbrace{\sum_{j=1}^{n} A_{ij}}_{\text{Degree of node i, } d_i}$$

$$= c\sum_{j=1}^{n} A_{ij}u_j - cu_i d_i$$

We now make use of the Kronecker delta, δ_{ij}

$$\delta_{ij} = \begin{cases} 0, \text{ if } i \neq j \\ 1, \text{if } i = j \end{cases}$$

Diffusion on a Graph

so $\quad cu_i d_i = c\sum_{j=1}^{n} \delta_{ij} u_j d_j$

$$\frac{du_i}{dt} = c\sum_{j=1}^{n} A_{ij}u_j - c\sum_{j=1}^{n} \delta_{ij}u_j d_j$$

Define the n-dimensional vector $\quad \vec{u} = \begin{pmatrix} u_1 \\ \cdots \\ u_n \end{pmatrix}$

u

Then $\quad c\sum_{j=1}^{n} A_{ij}u_j = c[A\vec{u}]_i$

Next define the $n \times n$ degree matrix $\quad D = \begin{bmatrix} d_1 & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & d_n \end{bmatrix}$

Then $\quad c\sum_{j=1}^{n} \delta_{ij}u_j d_j = c[D\vec{u}]_i$

so

$$\frac{du_i}{dt} = c\sum_{j=1}^{n} A_{ij}u_j - c\sum_{j=1}^{n} \delta_{ij}u_j d_j$$

becomes

$$\frac{d\vec{u}}{dt} = cA\vec{u} - cD\vec{u}$$

$$= c(A-D)\vec{u}$$

or

$$\frac{d\vec{u}}{dt} + c(D-A)\vec{u} = \vec{0}$$

We now define the Graph Laplacian matrix,

$$\boxed{L \equiv D - A}$$

$\shortparallel$

∇^2

The equation for diffusion on a graph is then

$$\frac{du}{dt} = \nabla^2 u \iff \boxed{\frac{d\vec{u}}{dt} + cL\vec{u} = \vec{0}}$$

The Graph Laplacian

What's inside of L?

$L -$ A

$$L_{ij} = \begin{cases} d_i \text{, if } i = j \\ -1 \text{, if } i \neq j \text{ and there is an edge} \\ 0 \text{, if } i \neq j \text{ and there is no edge} \end{cases}$$

Is L symmetric?   Yes, why?

Solving the Graph Diffusion Equation

$$\frac{d\vec{u}}{dt} + cL\vec{u} = \vec{0}$$

$u =$

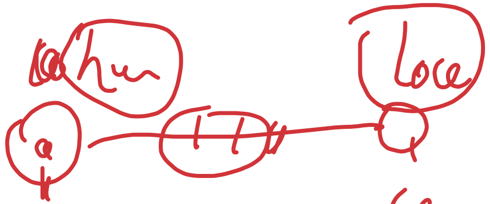
$u = e^{cLt}$

$\frac{du}{dt} =$ differ.

$f(x-\Delta x) \quad f(x) \quad f(x+\Delta x)$

to

$x-\Delta x \quad x \quad x+\Delta x$



u_j

$t_0 = \frac{du_i}{dt} = (u_i - u_j)^2$

Solving the Graph Diffusion Equation

f cos sin

$$\frac{d\vec{u}}{dt} + cL\vec{u} = \vec{0}$$

$Du = Lu$

This is a linear system of ODEs, so it is solvable. Also, since L is symmetric it has real eigenvalues and **orthogonal eigenvectors**, $\vec{v}_i$, i= $1, \cdots, n$.

Now write the solution as a linear combination of these eigenvectors, noting that the coefficients change over time: $\vec{u} = \sum_{i=1}^{n} a_i(t)\vec{v}_i$.

Why?

- **orthogonal "bases" of linear transform**

Insert this into the ODE, $\displaystyle\sum_{i=1}^{n} \frac{da_i}{dt}\vec{v}_i + \sum_{i=1}^{n} ca_i L\vec{v}_i = 0$

$$\Rightarrow \sum_{i=1}^{n} \left(\frac{da_i}{dt} + ca_i\lambda_i\right)\vec{v}_i = 0$$

$\vec{v}_i v_j \quad i \neq j = 0$

where λ_i is an eigenvalue of L.

Solving the Graph Diffusion Equation

Now take the inner product of both sides of the last equation with each of the eigenvectors, recalling that they form an orthogonal set. This leads to *n* differential equations for the coefficients $a_i(t)$.

$$\frac{da_i}{dt} + c\lambda_i a_i = 0 \text{ , } i\text{=1,…,n}$$

1-D

These ODEs are uncoupled and linear, so they have simple exponential solutions:

$$a_i(t) = a_i(0)e^{-c\lambda_i t}$$

L

t=0

u(0)

where $a_i(0)$ is the initial value of the coefficient.

Since each coefficient has such a solution, then by the superposition principle, a linear combination of these is also a solution. Thus, the general solution to the graph diffusion differential equation is

$$\boxed{\vec{u}(t) = \sum_{i=1}^{n} a_i(0)e^{-c\lambda_i t}\vec{v}_i}$$

Spectral Solution !

What we got so far

- **Diffusion function** on graph is a Linear ODE with trackable solutions parameterized by its **Laplacian mat (L)**
- **Laplacian mat (L)** is the **Laplacian operator on graph**

$$\Delta f = \nabla^2 f = \nabla \cdot \nabla f = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i^2}$$

Other ode on graphs

The **Laplace equation**

$$Lu = \lambda u$$

$\nabla u_{xx} = \lambda u$

the **Poisson equation**

$$Lu = g$$

Given an initial position and velocity, the **wave equation**

$$u_{tt} = -Lu$$

IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS, VOL. 3, NO. 3, SEPTEMBER 2017

Learning Heat Diffusion Graphs

Dorina Thanou, Xiaowen Dong, Daniel Kressner, and Pascal Frossard

Task: Graph structure/topology learning

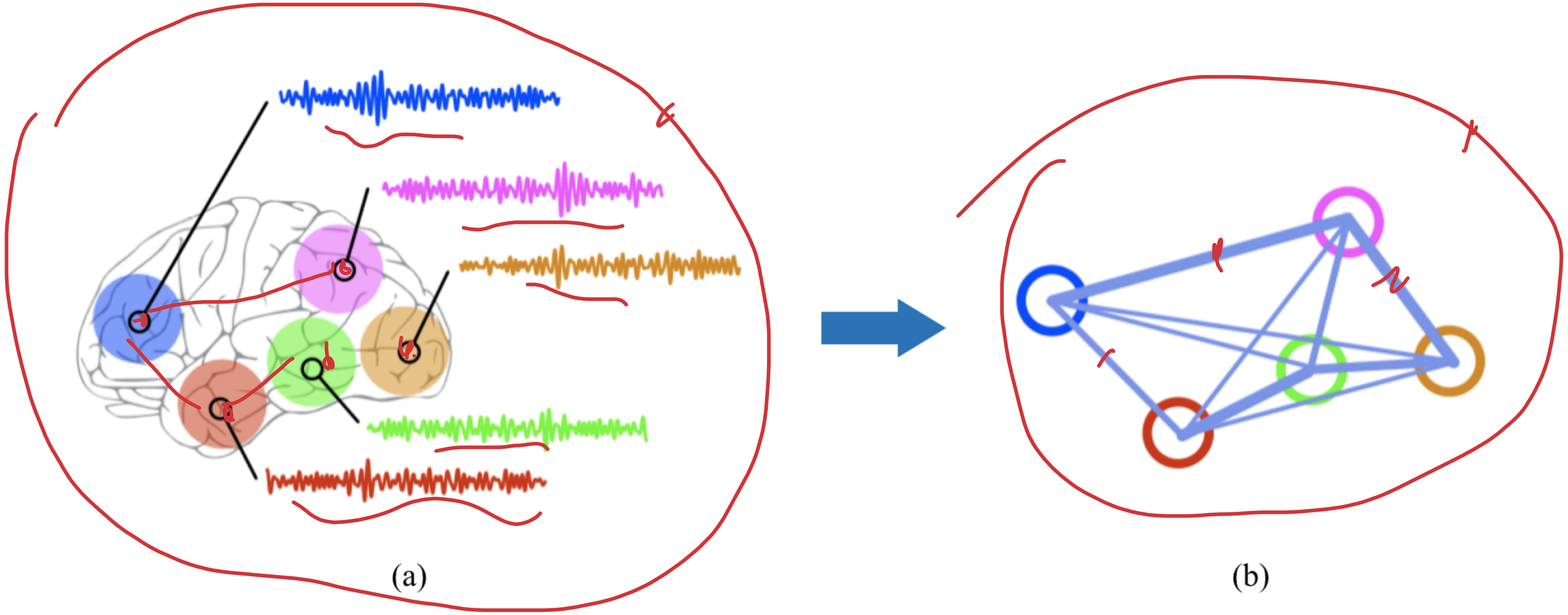
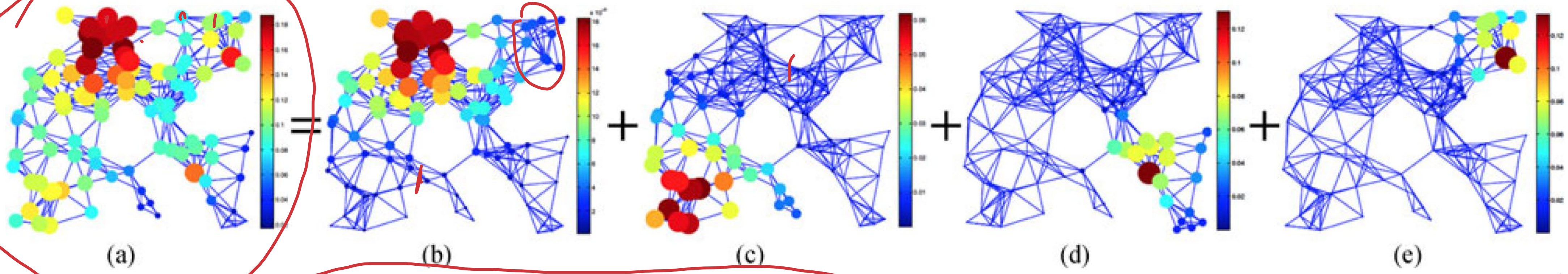


Fig. 1: Inferring functional connectivity between different regions of the brain. (a) BOLD time series recorded in different regions of the brain. (b) A functional connectivity graph where the vertices represent the brain regions and the edges (with thicker bars indicating heavier weights) represent the strength of functional connections between these regions. Figure adapted from [4] with permission.

- **Motivation: Dictionary learning of graph**



A graph signal is a function $x : \mathcal{V} \rightarrow \mathbb{R}$ such that $x(v)$ is the value of the function at the vertex $v \in \mathcal{V}$. We consider the factor analysis model from [15] as our graph signal model, which is a generic linear statistical model that aims at explaining observations of a given dimension from a set of unobserved latent variables. Specifically, we consider

$$x = \mathcal{D}h + \overset{=0}{u_x} + \epsilon, \tag{1}$$

where $x \in \mathbb{R}^N$ is the observed graph signal, $h \in \mathbb{R}^K$ is the latent variable that controls x, and $\mathcal{D} \in \mathbb{R}^{N \times K}$ is a representation matrix that linearly relates the two variables, with $K \geq N$. The parameter $u_x \in \mathbb{R}^N$ is the mean of x, which we set to zero for

$(V, \mathcal{E}, W)$

Dh

$N \times K \cdot = \in N$

How to define the "dictionary" of graph state

- Recall the **solution of diffusion function on Graph**

$$\frac{\partial x}{\partial \tau} - Lx = 0,\ x(v,0) = x_0(v) \tag{5}$$

t

where $x(v, \tau)$ describes the heat at node v at time τ, beginning from an initial distribution of heat given by $x_0(v)$ at time zero. The solution of the differential equation is given by

Different \tau or t define different graph state!

$$x(v,\tau) = e^{-\tau L} x_0(v). \tag{6}$$

$= \sum a_0(t) \cdot V_i$

Going back to our graph signal model, the graph heat diffusion operator is defined as [20]

$$\widehat{g}(L) = e^{-\tau L} = \chi e^{-\tau \Lambda} \chi^T.$$

Dictionary of graph spectrum: set different t

graph. Specifically, one can consider spectral graph dictionaries defined by filtering the eigenvalues of the graph Laplacian in the following way:

$\tau = 1, \cdots S$

$$\mathcal{D} = [\widehat{g}_1(\mathcal{L})\ \widehat{g}_2(\mathcal{L}) \ldots \widehat{g}_S(\mathcal{L})], \tag{2}$$

where $\{\widehat{g}_s(\cdot)\}_{s=1,\ldots,S}$ are graph filter functions defined on a domain containing the spectrum of the graph Laplacian. Each of these filters captures different spectral characteristics of the graph signals.

$$x = \mathcal{D}h + \epsilon = [e^{-\tau_1 L}\ e^{-\tau_2 L}\ \cdots\ e^{-\tau_S L}]h + \epsilon,$$

$$x = \sum_{s=1}^{S} e^{-\tau_s L} h_s.$$

Given a set of M signal observations $X = [x_1, x_2, \ldots, x_M] \in \mathbb{R}^{N \times M}$, resulting from heat diffusion processes evolving on an unknown weighted graph $\mathcal{G}$, our objective is twofold: (i) infer the graph of N nodes by learning the graph Laplacian L, and (ii) learn, for each signal, the latent variable that reveals the sources of the observed processes, i.e., $H = [h_1, h_2, \ldots, h_M]$ and the diffusion parameters $\tau = [\tau_1, \tau_2, \ldots, \tau_S]$. As the graph Laplacian L captures the sparsity pattern of the graph, learning L is equivalent[2] to learning the graph $\mathcal{G}$. This results in the following joint optimization problem for H, L, and τ:

$$
\begin{aligned}
\underset{L,H,\tau}{\text{minimize}} \quad & \|X - \mathcal{D}H\|_F^2 + \alpha \sum_{m=1}^{M} \|h_m\|_1 + \beta \|L\|_F^2 \\
\text{subject to} \quad & \mathcal{D} = [e^{-\tau_1 L}\; e^{-\tau_2 L} \ldots e^{-\tau_S L}] \\
& \operatorname{tr}(L) = N, \\
& L_{ij} = L_{ji} \le 0, \; i \neq j, \\
& L \cdot \mathbf{1} = \mathbf{0}, \\
& \tau \ge 0,
\end{aligned} \qquad (7)
$$

where h_m corresponds to the mth column of the matrix H. Ac-

M node

N

M node h

w

A

A-D

L

Algorithm 1: Learning heat kernel graphs (**LearnHeat**).

Input: Signal set X, number of iterations iter

Output: Sparse signal representations H, graph Laplacian L, diffusion parameter τ

Initialization: $L = L^0$, $\mathcal{D}^0 = [e^{-\tau_1 L}\, e^{-\tau_2 L} \ldots e^{-\tau_S L}]$, $\tau = \tau^0$

for $t = 1, 2, \ldots, \mathsf{iter}$ **do:**

  Choose $c_t = \gamma_1 C_1(L^t, \tau^t)$

    Update H^{t+1} by solving opt. problem (9)

  Choose $d_t = \gamma_2 C_2(H^{t+1}, \tau^t)$

   (a) Update L^{t+1} by solving opt. problem (11)

   (b) Update $\mathcal{D}^{t+1} = [e^{-\tau_1^t L^{t+1}} \ldots e^{-\tau_S^t L^{t+1}}]$

  Choose $e_t = \gamma_3 C_3(L^{t+1}, H^{t+1})$

   (a) Update τ^{t+1} by solving opt. problem (12)

   (b) Update $\mathcal{D}^{t+1} = [e^{-\tau_1^{t+1} L^{t+1}} \ldots e^{-\tau_S^{t+1} L^{t+1}}]$

end for

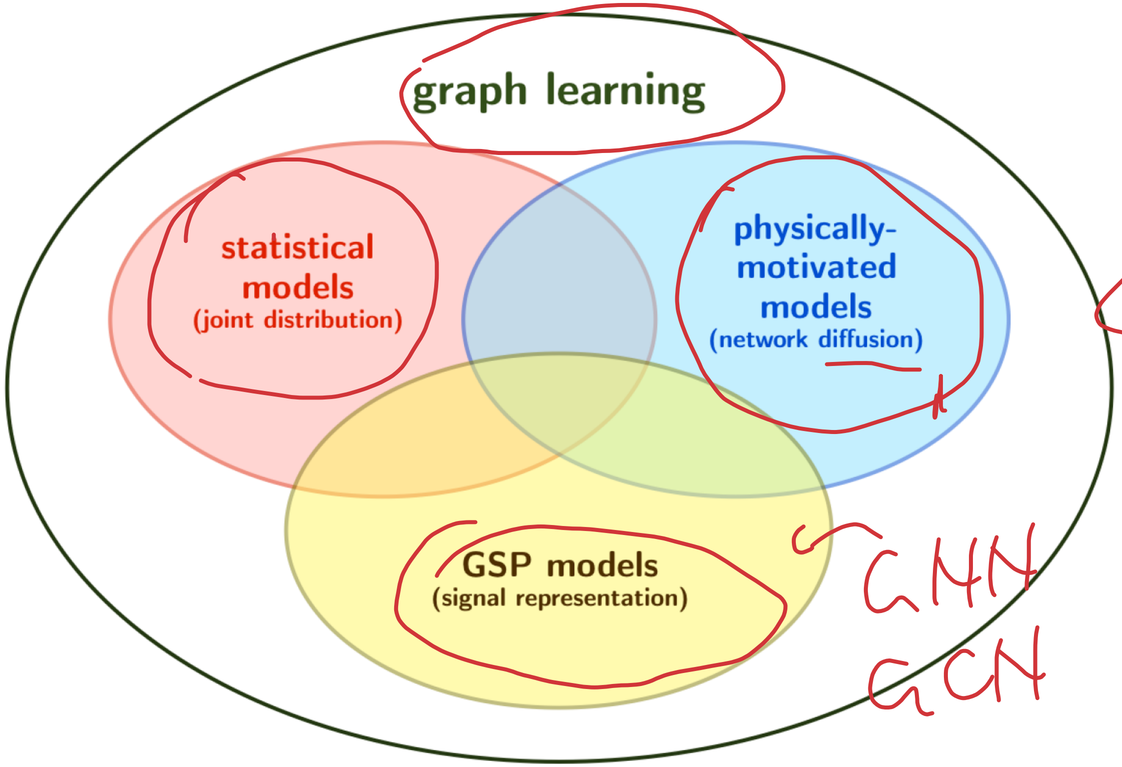
$L = L^{\mathsf{iter}}, H = H^{\mathsf{iter}}, \tau = \tau^{\mathsf{iter}}$.

200

20~~0~~000

Other topics of graph learning

S ODE on g



L

GNN
GCN

Fig. 2: A broad categorization of different approaches to the problem of graph learning.

See more details at: Dong, Xiaowen, et al. "Learning graphs from data: A signal representation perspective." *IEEE Signal Processing Magazine* 36.3 (2019): 44-63.