CS07 Research methods in Complex system

Bingsheng Chen

October 5, 2023

1 Introduction

In complex system topics, one of the research directions is to use the observed data in order to validate the mathematical model built to understand the mechanism [1]. However, analyzing data is not trivial since it involves the techniques of inference. In these lecture notes, we introduce basic terminologies and commonly used methods in inference. Furthermore, some advanced applications of inference in complex systems are summarized to help us better understand inference methods, including inferencing parameters in stochastic processes and understanding high dimensional interactions in neural networks.

In the first section, we discuss the parametric Bayesian inference method with the examples of curve fitting problems. Then non-parametric methods, Gaussian process, were introduced to solve the linear regression problem, meanwhile, the method is extended to infer the drift term of Stochastic Differential Equation from a dense observation. This approach gives a inaccurate inference result in the sparse observations, therefore, E-M algorithm(latent-variable approach) was used to obtain a better result. Meanwhile, to overcome the posterior distribution is too complex to allow for analytical computations when the number of variables is large, approximate inference methods were needed. We specifically illustrate how the mean-field approximation can be used to approximate an intractable probability distribution. In the second section, we discussed how different types of stochastic block models can be used to perform inference on complex networks. We specifically focus on missing link problems and compare the result of stochastic block models to other method. At the end of section, we discussed the different criterion for model selection, including Bayesian Information Criterion to regularize the parameters. In the third section, we discuss why the classical asymptotic inference can not apply to high-dimensional inference. Several methods, including PCA, auto-encoding and restricted Boltzmann machine, were introduced that can be used to compress the data from high-dimension to low-dimension representation to improve inference result.

2 Tractable approximate inference for stochastic processes

2.1 Introduction and parametric method

The key goal of inference problems is that given a set of data x, what will the most likely y outcome be. Let us defined our variables properly. Firstly, we denotes k observations from a data set as a vector, so that $\mathbf{y} \equiv (y_1, \dots, y_K)$ and latent variable as $\mathbf{z} \equiv (z_1, \dots z_N)$. Latent variable is not an observable variable from the data set directly but it can to used to explain the observation from the data, which can be demonstrated in a curve fitting problem[2]. We define $\mathbf{y}$ as the training set used to produce a fitted curve and $\mathbf{z}$ is the predicted outcome(latent variable) given by the fitted curve. There might be some uncertainties in our measurements, therefore, we define likelihood function $p(\mathbf{y}|\mathbf{z})$ which denotes the probability of data been observed $\mathbf{y}$ if we fixed set of latent variable $\mathbf{z}$. The model is known as a forward model. Furthermore, we will have some beliefs(bias) on our latent variables before we have seen the data, where we define it as *prior information*, $p(\mathbf{z})$. From Bayes' rule, we will have a probability(posteriori probability) of a set of latent variable $\mathbf{z}$ if a set of observations were given:

$$p(\mathbf{z}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{y})}. \tag{1}$$

We can derive the marginal distribution $p(z_i|\text{data})$ from posterior distribution as following:

$$p(z_i|\text{data}) = \int \frac{p(\text{data}|z_1, \dots, z_N)p(z_1, \dots, z_N)}{p(\text{data})} dz_1 \cdots dz_{i-1} dz_{i+1} \dots dz_N, \tag{2}$$

and also the evidence function $p(\text{data})$(i.e. the probability of the data being observed across all models),

$$p(\text{data}) = \int p(\text{data}|z_1, \ldots, z_N)p(z_1, \ldots, z_N)dz_1 \ldots dz_N, \tag{3}$$

For a generative model, we can predict the probability of a observation y if an input x and the model is given, for instance, consider a simplistic curve fitting problem, a observation is explained by a deterministic fitted result plus an random noise,

$$y_i = f_\theta(x_i) + v_i, \tag{4}$$

where v_i is independent Gaussian noise with zero mean and variance σ^2, where $f_\theta(x_i)$ is a function with unknown parameters θ that maps independent variable x_i to an explanatory variable. For a linear line fitting problem, we have $f_\theta(\mathbf{x}) = \theta\mathbf{x}$, the choices of $f_\theta(\mathbf{x})$ are not unique, it can also be a combination of non-linear function, therefore we can generalize the function,

$$f_\theta(\mathbf{x}) = \sum_{l=1}^{K} \theta_l \phi_l(\mathbf{x}), \tag{5}$$

where $\phi_l(\mathbf{x})$ is different for each scenario. For instance, if the parametric function is a power series then $\phi_l(\mathbf{x}) = x^l$. Choosing too many parameters is sometimes not easy to handle and also causes the problem of over-fitting[3], i.e. higher degree polynomials initially over fit, therefore, we need to control the number of parameters K, the detailed method will be discussed in model selection.

If the data points are independent to each other, we can write the likelihood function $p(\mathbf{y}|\theta)$ as a joint distribution:

$$p(\mathbf{y}|\theta) = \prod_{i=1}^{N} p(y_i|\theta) \propto \exp\left(-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{f}_\theta(\mathbf{x}))^\mathsf{T}(\mathbf{y} - \mathbf{f}_\theta(\mathbf{x}))\right). \tag{6}$$

From Equation 6, we can observe that the likelihood is a Gaussian density function. Then if $p(\theta)$ is also a Gaussian prior, the posterior $p(\theta|\mathbf{y})$ will also be a Gaussian density function. Before discussing how the parametric method can be used in inference, we need to introduce some properties about the Gaussian distribution since we noticed the likelihood function is Gaussian from Equation (6) . Let $(\mathbf{x}, \mathbf{y})^\mathsf{T}$ be 2-d multivariate normal variables with a given covariance matrix Σ(we also denotes Ω as the inverse of Σ, i.e. $\Omega = \Sigma^{-1}$). Notice, the covariance matrix Σ has to be positive-definite [2], which means for any vector $\mathbf{a} \neq 0$, $\mathbf{a}^\mathsf{T}\Sigma\mathbf{a} > 0$. The joint distribution of $\mathbf{x}$ and $\mathbf{y}$ can be written as:

$$p(x, y) \propto \exp\left\{-\frac{1}{2}\left[(\mathbf{x}, \mathbf{y})^\mathsf{T}\Omega(\mathbf{x}, \mathbf{y}) + (\mathbf{x}, \mathbf{y})^\mathsf{T}\xi\right]\right\}. \tag{7}$$

The mean μ of multivariate normal variable x and y can be derived by the fact that the Gaussian density is uniquely peaked at the mean. Denoting $(\mathbf{x}, \mathbf{y}) = \mathbf{z}$, taking the derivative to find the maximum point, we find $\mu = \mathbf{z}_{\text{ML}} = \Omega^{-1}\xi$.

Now if we consider the marginal distribution $p(x|y)$, since $p(y)$ is a given condition so it is treated as a constant. So we only need to consider term remaining with x, we can invoke that $p(\mathbf{y}|\mathbf{x}) \propto p(\mathbf{x}, \mathbf{y})$, i.e.

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{y})} \propto \exp\left\{-\frac{1}{2}\left[\mathbf{x}^\mathsf{T}\Omega_{xx}\mathbf{x} + \mathbf{x}^\mathsf{T}(\xi - \Omega_{xy})\right]\right\}. \tag{8}$$

We can compute the conditional mean $\mathbb{E}(x|y)$ by either completing the square or implying the method of determining maximum likelihood of $\mathbf{x}$ as above $\mathbf{x}_{ML}$ by setting $\xi = 0$. We find that $\mathbb{E}(x|y) = \mathbf{x}_{\text{ML}} = -(\Omega_{xx})^{-1}\Omega_{xy}$. We find the covariance $\Sigma_{xx|y} = (\Omega_{xx|y})^{-1} = (\Omega_{xx})^{-1}$ by rewriting the conditional probability as:

$$p(\mathbf{x}|\mathbf{y}) \propto \exp((\mathbf{x} - \mu(\mathbf{x}|\mathbf{y}))^\mathsf{T}\boldsymbol{\Sigma}_{\mathbf{xx}|\mathbf{y}}(\mathbf{x} - \mu(\mathbf{x}|\mathbf{y})). \tag{9}$$

2.2 A note on Prior Choosing

In principle, we should always choose the prior that maximize entropy[10]. Loosely speaking, maximize the entropy means we will have more uncertainty in our previous knowledge and therefore we can gain more knowledge about the original distribution from observations. This is generally achieved by solving the Lagrange multiplier, subject to some constrains and as a consequence, we will obtain an exponential distributed prior. The drawback of using these families of prior is that while calculating the marginal distribution, the posterior is not always integrable. Moreover, if we have enough observations, the prior will become less relevant. Therefore, we generally choose conjugate prior i.e. the prior have the same distribution family of the posterior while performing inference.

2.3 Gaussian Process

Gaussian Process is a very powerful non-parametric method with explicit uncertainty that are commonly used in classification and regression problems[4]. It is non-parametric because instead of fitting parameters with using a selected model, the kernel method uses the measured similarities between observations, based on correlation length, to predict an outcome for a new input[5]. The type of correlation is controlled by kernel function.

Definition 1. Gaussian Process *is a collection of random variables and the joint distribution of any of its subsets are also Gaussian distribution.*

The Gaussian process is fully characterized by its mean and covariance. Consider a real process $f(\mathbf{x})$, with mean function of $(m(\mathbf{x}))$ and the covariance function of $k(\mathbf{x}, \mathbf{x}')$. If the process is Gaussian process, then we can denote $f(x)$ as:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \tag{10}$$

For simplicity we choose our $m(\mathbf{x}) = 0$ and the covariance kernel can be written as

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}(f((\mathbf{x})f(\mathbf{x}')). \tag{11}$$

The Kernel choice is not unique and dependent on our belief, however, a valid kernel need to be positive definite. Here we show a technique that uses Fourier Transform to show the kernel is positive definite. If a kernel $K(\mathbf{x} - \mathbf{x}')$is stationary(i.e. translational invariance $K(x, x') = K|x - x'|$) and can be written as the Fourier representation as below:

$$K(x) = \int_{\infty}^{\infty} e^{i\omega x} \hat{K}(\omega) d\omega, \tag{12}$$

then the kernel is positive-definite only if the Fourier transform of kernel $\hat{K}(\omega)$ is positive. The proof is sketched as following:

$$\begin{aligned} \mathbf{a}^{\mathsf{T}} K \mathbf{a} = \sum_{i,j} a_i a_j K(x_i, x_j) &= \sum_{i,j} a_i a_j \int_{-\infty}^{\infty} e^{-i\omega(x_i - x_j)} \hat{K}(\omega) d\omega \\ &= \int \left(\sum_i a_i e^{\mathbf{i}\omega x_i} \right) \left(\sum_j a_j e^{-\mathbf{i}\omega x_j} \right) \hat{K}(\omega) d\omega. \end{aligned} \tag{13}$$

where i, j are dummy variables, so they are interchangeable and also we know that the summation of i and j are complex conjugate, therefore, we have:

$$\mathbf{a}^{\mathsf{T}} K \mathbf{a} = \int \left| \sum_i a_i e^{\mathbf{i}\omega x_i} \right|^2 \hat{K}(\omega) d\omega > 0 \qquad \forall a \neq 0. \tag{14}$$

There are many different valid kernels can be used based on scenarios, one of the most popular kernel is called *Radial Basis function*(RBM), where $k(\mathbf{x}, \mathbf{x}') = \exp(-\lambda|\mathbf{x} - \mathbf{x}'|^2)$. *Ornstein-Uhlenbeck*(OU) kernel is also valid kernel, where $k(\mathbf{x}, \mathbf{x}') = \exp(-\lambda|\mathbf{x} - \mathbf{x}'|)$, where λ is called hyper-parameter. RBM treat the correlation between two observables are normally distributed, where the correlation length scale is dependent on $|\mathbf{x} - \mathbf{x}'|^2$. However, RBM is not ideal in modeling stochastic multivariate variable, OU kernel gives a shorter correlation length so the discontinuity in Brownian motion can be simulated. The choice of hyper-parameter is generally determined by maximizing the evident function $p(\mathbf{y})$, an example in Gaussian Regression will be given in the following section. Furthermore, the linear operation(differentiation) on kernel is also valid kernel, which can be shown from the first principle, in this case we choose the covariance kernel as the kernel we used, start from $\mathbb{E}[f(x_1)f'(x_2)]$:

$$\mathbb{E}[f(x_1)f'(x_2)] = \lim_{\epsilon \to 0} \frac{1}{\epsilon} \mathbb{E}[f(x_1)(f(x_2 + \epsilon) - f(x_2))] = \lim_{\epsilon \to 0} \frac{1}{\epsilon} \mathbb{E}[K(x_1, x_2 + \epsilon) - K(x_1, x_2)] = \frac{\partial}{\partial x_2} K(x_1, x_2). \tag{15}$$

This result is important if we are performing inference for dynamical system, i.e. Linear ODE.

2.4 Gaussian Regression

Let us explore a regression example of forward model to see how GP can be used for inference. Firstly, we assume each observation y_i is sampled from a Gaussian distribution, which is now represented by the "true value"(inferred result without noise) of the observation $f(x_i)$ plus some additive independent Gaussian noise as follows:

$$y_i = f(x_i) + v_i, \qquad f(\mathbf{x}) \sim \mathcal{GP}(0, K), \tag{16}$$

where y_i is the i-th observed data point , v_i is a Gaussian noise have zero mean and σ^2 variance and Covariance kernel K. For simplicity, we denotes $f(x_i)$ as z_i. If we have a new input vector $\mathbf{x}^*$ and a prediction of $f(\mathbf{x}^*)$, we can calculate how likely of this data is going to be predicted i.e.the predictive distribution $p(f(x*)|\mathbf{y})$ as follow:

$$p(f(\mathbf{x}^*)|\mathbf{y}) = \int p(f(\mathbf{x}^*), \mathbf{z}|\mathbf{y}) d\mathbf{z} = \int p(f(\mathbf{x}^*)|\mathbf{z}, \mathbf{y}) p(\mathbf{z}|\mathbf{y}) d\mathbf{z}. \tag{17}$$

Since the latent variable already contained informations from the data, we can eliminate the dependencies that $p(f(\mathbf{x}^*)|\mathbf{z}, \mathbf{y}) = p(f(\mathbf{x}^*)|\mathbf{z})$. Use Gaussian prior $p(\mathbf{z}) \propto \exp(\mathbf{z}^\mathsf{T}\mathbf{K}^{-\mathbf{1}}\mathbf{z})$ and also the likelihood, we can derive posterior from Bayes theorem, i.e.

$$p(\mathbf{z}|\mathbf{y}) = \frac{p(\mathbf{z}|\mathbf{y})p(\mathbf{z})}{p(\mathbf{y})} \propto e^{-\frac{1}{2\sigma^2}\mathbf{z}^\mathsf{T}\mathbf{z} + \frac{1}{\sigma^2}\mathbf{z}^\mathsf{T}\mathbf{y} - \mathbf{z}^\mathsf{T}\mathbf{K}^{-\mathbf{1}}\mathbf{z}}. \tag{18}$$

Re-arrange the equation, use the same approach we used at the first section and we found our posterior is another Gaussian distribution,i.e. $p(\mathbf{z}|\mathbf{y}) \sim \mathcal{N}(\mathbf{z}|\mu, \mathbf{S})$, where $\mathbf{S} = \mathbf{K}^{-\mathbf{1}} + (1/\sigma^2)\mathbf{I}$ and $\mu = (1/\sigma^2)\mathbf{S}\mathbf{y}$. Before calculating the predictive distribution, we need to calculate $p(f(\mathbf{x}^*)|\mathbf{z})$. From the result of conditional Gaussian, we know that $p(f(\mathbf{x}^*)|\mathbf{z}) \sim \mathcal{N}(f(\mathbf{x}^*)|\mathbf{m}, \mathbf{s})$, where $\mathbf{m} = -\mathbf{\Omega}^{-\mathbf{1}}_{\mathbf{x}^*\mathbf{x}^*}\mathbf{\Omega}_{\mathbf{x}^*\mathbf{x}}\mathbf{z}$ and $\mathbf{s} = -\mathbf{\Omega}^{-\mathbf{1}}_{\mathbf{x}^*\mathbf{x}^*}$.We can use the kernel and recall that the correlation between two observations only depends on the choice of kernel function and relative distance. The covariance matrix $\mathbf{\Sigma}$ can be written as,

$$\Omega = \mathbf{\Sigma}^{-1} = \begin{pmatrix} \mathbf{K}(\mathbf{x}^*, \mathbf{x}^*) & \mathbf{K}(\mathbf{x}, \mathbf{x}^*) \\ \mathbf{K}(\mathbf{x}^*, \mathbf{x}) & \mathbf{K}(\mathbf{x}, \mathbf{x}) \end{pmatrix}^{-1}. \tag{19}$$

Again $\mathbf{x}$ is the training set. We abbreviate $\mathbf{K}(\mathbf{x}, \mathbf{x})$ as $\mathbf{K}$. In the case, we only consider one new test point case and we denotes $\mathbf{k_x}$ as covariance between new test point and n-training points, i.e. $\mathbf{k_x} = (K(x^*, x_1), K(x^*, x_2), \ldots, K(x^*, x_n))^\mathsf{T}$. Use the inverse block matrix result to obtain ${\mathbf{\Omega}_{\mathbf{x}^*\mathbf{x}^*}}^{-1}$ and $\mathbf{\Omega}_{\mathbf{x}^*\mathbf{x}}$, we can obtain conditional mean $\mathbf{m}$ and covariance $\mathbf{s}$ as below:

$$m = \mathbb{E}(f(x)|\mathbf{z}) = {\mathbf{k_x}}^\mathsf{T}\mathbf{K}^{-1}\mathbf{z}, \qquad \mathbf{s} = \mathrm{VAR}[f(x)|\mathbf{z}] = \mathrm{K}(x^*, x^*) - {\mathbf{k_x}}^\mathsf{T}\mathbf{K}^{-1}\mathbf{k_x}. \tag{20}$$

Therefore, mean outcome of prediction for the regression problem can be computed for the new input point with a given training set as:

$$\begin{aligned} \mathbb{E}[f(x^*)|\mathbf{y}] &= \int \mathbb{E}(f(x)|\mathbf{z}) p(\mathbf{z}|\mathbf{y}) d\mathbf{z} = {\mathbf{k_x}}^\mathsf{T}\mathbf{K}^{-1} \int \mathbf{z} p(\mathbf{z}|\mathbf{y}) d\mathbf{z} = {\mathbf{k_x}}^\mathsf{T}\,\mathbb{E}(\mathbf{z}|\mathbf{y}) \\ &= \frac{1}{\sigma^2}{\mathbf{k_x}}^\mathsf{T}\mathbf{S}\mathbf{y} = \frac{1}{\sigma^2}{\mathbf{k_x}}^\mathsf{T}\left(\mathbf{K} + \frac{1}{\sigma^2}\mathbf{I}\right)^{-1}\mathbf{y} = {\mathbf{k_x}}^\mathsf{T}\left(\mathbf{K} + \sigma^2\mathbf{I}\right)^{-1}\mathbf{y}. \end{aligned} \tag{21}$$

We can also derive that the uncertainty(variance) of prediction is $\mathrm{VAR}[f(x^*)|\mathbf{y}] = \mathrm{K}(x^*, x^*) - {\mathbf{k_x}}^\mathsf{T}\mathbf{K}^{-1}\mathbf{k_x}$. This is informative, we can certainly notice that the variance is independent of training set $\mathbf{y}$ we use. Furthermore, since the kernel is positive definite, the greater size of training set will give a more precise predictions .

We need to determine sensible hyper-parameters and noise used in kernel as promised. We can determine these parameters via using Automatic Relevance Determination(ARD) method. ARD method numerically searches for best-fit hyper-parameters that optimize the evidence function(i.e. $p(\mathbf{y})$) :

$$p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{z}) p(\mathbf{z}) d\mathbf{z} = \frac{1}{(2\pi)^{n/2}\left|\det(\mathbf{K} + \sigma^{\mathbf{2}}\mathbf{I})\right|^{1/2}} \exp\left[\frac{1}{2}\mathbf{y}^\mathsf{T}(\mathbf{K} + \sigma^{\mathbf{2}}\mathbf{I})^{-1}\mathbf{y}\right]. \tag{22}$$

2.5 SDE and inference application of GP process in SDE

In the experiment, we will have an uncertainty associate with the differential equation, in order to account this, we simulate the dynamical system with noise via stochastic differential equation:

$$dX_t = \underbrace{f(X_t)}_{\text{Drift}} dt + \underbrace{D^{\frac{1}{2}}(X_t)}_{\text{Diffusion}} dW_t. \tag{23}$$

The drift term correspond to a deterministic driving force in the language of physics and W_t is Brownian motion, i.e. $dW_t \sim N(0, dt)$. In computer, we can simulate the above stochastic process in the limit of discrete time as:

$$X_{t+\Delta t} - X_t = \underbrace{f(X_t)}_{\text{Drift}} \Delta t + \underbrace{D^{\frac{1}{2}}(X_t)}_{\text{Diffusion}} \sqrt{\Delta t} \epsilon_t. \tag{24}$$

where ϵ is a standard normal. We are interested in average increment(conditional expectation), which can be used to give a prediction for future evolution, i.e.$\mathbb{E}[(x_{t+\Delta t} - x_t)/\Delta t | x_t = X] = f(X_t)$, since the increment of Brownian motion is i.i.d with mean of zero. Similarly, the diffusion constant D can be computed from the variance of the increment:

$$\begin{aligned} D(X_t) &= \frac{1}{\Delta t} \lim_{\Delta t \to 0} \mathbb{E}\,\mathrm{Var}[(x_{t+\Delta t} - x_t) | x_t = X] \\ &= \lim_{\Delta t \to 0} \left\{ \frac{1}{\Delta t} \mathbb{E}\left[(x_{t+\Delta t} - x_t)^2 | x_t = x\right] - \underbrace{\mathbb{E}\left[(x_{t+\Delta t} - x_t) | X_t = x\right]^2}_{=0} \right\} \\ &= \lim_{\Delta t \to 0} \frac{1}{\Delta t} \mathbb{E}\left[(x_{t+\Delta t} - x_t)^2 | x_t = x\right]. \end{aligned} \tag{25}$$

Now we assume diffusion matrix D is independent on X_t and a diagonal matrix so that the paths in each dimension are independent . Suppose we observed a path of n-dimensional observations over the time interval $[0, T]$ with each time partition $\Delta t \to 0$, the transitional probabilities of the process are Gaussian, as:

$$p_f(X_{0:T}|\mathbf{f}) \propto \underbrace{\exp\left[\frac{1}{2\Delta t} \sum_t \|X_{t+\Delta t} - X_t\|^2\right]}_{p_0(X)} \times \underbrace{\exp\left[\frac{1}{2} \sum_t \|f(X_t)^2\| \Delta t + \sum_t f(X_t) \cdot (X_{t+\delta t} - X_t)\right]}_{L(X_{0:T}|f)}. \tag{26}$$

To obtain a non-parametric estimate of the drift $f(x)$, we noticeable that $L(X_{0:T}|f)$ contains the drift f quadratically therefore, if we use the conjugate prior for $f(x)$ as a Gaussian process, i.e. $\mathbf{f} \sim P_0(\mathbf{f}) = GP(0, K)$, where K is a kernel, the posterior is also Gaussian. We restrict each component(dimension) of the drift is independent and we assume that diagonal diffusion matrix $D = \mathrm{diag}(\sigma_1^2, \cdots, \sigma_d^2)$. In this case we can estimate drift exponents by using ordinary GP regression with the training set, $\mathbf{d}^j = ((X^j_{t+\Delta t}) - X^j_t)^\mathsf{T}$ and kernel matrix $\mathbf{K}^j = K^j(X_s, X_t)$ and a test set $\mathbf{k}^j(x) = K^j(x, X_t)^\mathsf{T}$. With the method introduced in Gaussian regression, we can compute the mean of drift functions over posterior probability and also the variance of the estimations as following:

$$\tilde{f}^j(x) = \mathbf{k}^\mathsf{j}(x)^\mathsf{T} \left(\mathbf{K}^j + \frac{\sigma_j^2}{\Delta t}\mathbf{I}\right)^{-1} \mathbf{d}^j, \qquad \sigma^2_{(f^j)}(x) = K^j(x,x) - \mathbf{k}^j(x)\mathsf{t} \left(\mathbf{K}^j + \frac{\sigma_j^2}{\Delta t}\mathbf{I}\right)^{-1} \mathbf{k}^j(x). \tag{27}$$

However, this requires dense amount of data(i.e. $\Delta t \to 0$) and can not be applied when the observation is sparse. We can use variational approach to solve the sparse data problem. Treat process X_t between consecutive observations $k\tau < t < (k+1)\tau$ as hidden stochastic process with conditional path given by:

$$p(X_{0:T}|\mathbf{y}, f) \propto p(X_{0:T}|f) \prod_{k=1}^{n} \delta(y_k - X_{k\tau}), \tag{28}$$

where $\mathbf{y}$ is the collection of the observations y_k and we can use an EM algorithm to compute MAP prediction for the drift. EM algorithm is an iteration between the following two steps:

1. *Expectation Steps*: Compute the expected negative log of the likelihood function from full data.

$$\mathcal{L}(\mathbf{f}, q) = -\mathbb{E}_q[\ln L(X_{0:T}|\mathbf{f})] \tag{29}$$

   where q is a measure over paths that approximates the posterior $p(X_{0:T}|\mathbf{y}, \mathbf{f}_{\text{old}})$ from previous estimate.

2. Maximization Steps: Recompute the drift function as

$$f_{\text{new}} = \arg\min_{\mathbf{f}} (\mathcal{L}(\mathbf{f}, q) - \ln P_0(\mathbf{f})) \tag{30}$$

   where $P_0(\mathbf{f})$ is the same kernel we introduced above.

While we are computing $\mathcal{L}(f,q)$ at E-step, we approximate the discrete sum as a integral as follows:

$$\begin{aligned} -\mathbb{E}_q[\ln L(X_{0:T}|\mathbf{f})] &= \lim_{\Delta t\to 0}\frac{1}{2}\sum_t \mathbb{E}_q\left[\|f(X_t)\|^2\Delta t - 2\sum_t f(X_t)\cdot(X_{t+\Delta t}-X_t)\right] \\ &= \frac{1}{2}\int_0^T \mathbb{E}_q\left[\|f(X_t)\|^2 2\sum_t f(X_t)\cdot(X_{t+\Delta t}-X_t)\right]dt \\ &= \frac{1}{2}\int[\|f(x)\|^2 A(x)dx - \int f(x)\cdot y(x)dx, \end{aligned} \tag{31}$$

where q_t is the marginal probability of X_t computed from latent distribution q. Therefore, we can define approximated posterior drift:

$$g_t(x) = \lim_{\Delta t\to 0}\mathbb{E}_q[X_{t+\Delta t}-X_t|X_t=x], \tag{32}$$

where the function $A(x)$ and $y(x)$ is defined as:

$$A(x) = \int_0^T q_t(x)dt, \qquad \text{and} \qquad y(x) = \int_0^T g_t(x)q_t(x)dt. \tag{33}$$

Performing the iteration at EM steps, the result will converge maximum to local posterior. The method can also be used for other dynamical processes, such as Langevin dynamics, i.e. Port-Car problem[6].

2.6 Pseudo-Likelihood

Pseudo-likelihood was firstly introduced by Julian Besag, which use an alternative likelihood to approximates the original function[7]. In the binary-classification problem or non-Gaussian noise scenarios, original joint likelihood function need some approximations and GP is generally used as a latent-function. For instance, instead of modeling the data and assume the correlations between data is fully Gaussian:

$$p(z_1,\ldots,z_N|\text{data}) = \prod_{i=1}^n p(y_i|z_i)\exp\left[-\frac{1}{2}\mathbf{z}^\top\mathbf{K}^{-1}\mathbf{z}\right]. \tag{34}$$

we can approximate the likelihood with the Boltzmann correlation similar to neural network with topology of A:

$$p(z_1,\ldots,z_N|\text{data}) = \prod_i \psi_{z_i}\exp\sum_{i<j}A_{ij}z_iz_j. \tag{35}$$

2.7 Kullback–Leibler divergence

K-L divergence is a difference measure between two distributions. For two distributions $q(z)$ and $p(z)$, K-L divergence is defined as:

$$D(q||p) = \int q(z)\ln\frac{q(z)}{p(z)}dz. \tag{36}$$

From the Jensen's Equality we can illustrate it is always positive, however, if two distributions are identical as $q=p$, then K-L divergence will be zero. By rewriting the posterior $p(\mathbf{z}|\mathbf{y}) = \frac{p(\mathbf{y},\mathbf{z})}{p(\mathbf{z})}$, we can derived that:

$$p(\mathbf{z}|\mathbf{y}) = \arg\min_q\left[\mathbb{E}_q\left[\ln\frac{q(z)}{p(z,y)}\right]\right], \tag{37}$$

and the minimum K-L divergence is $-\ln p(y)$.

2.8 Generalized Mean-field approximation from minimizing K-L divergence

We can approximate the posterior $p(\mathbf{z}|\mathbf{y})$ with some simpler family of distribution so that the joint distribution can be factorized i.e.

$$p(\mathbf{z}|y) \approx q(\mathbf{z}) = \prod_{i=1}^n q_i(z_i). \tag{38}$$

We find the best approximation for q by minimizing K-L divergence, i.e.,

$$q^{\text{opt}}(\mathbf{z}) = \arg\min_{q\in\mathcal{F}} \left\{ \mathbb{E}_q \left[\frac{q(\mathbf{z})}{p(\mathbf{z},\mathbf{y})} \right] \right\} \tag{39}$$

Since each q is independent to each other according to Equation 38, as long as we know all other $q_{j\neq i}$, our integral can be split to summation of log term, which will only have a term left that depend on q_i(all other term is constant and independence of each z will lead to $\int q_i(z_i)dz_i = 1$), i.e.:

$$\begin{aligned}
\mathbb{E}_q \left[\frac{q(\mathbf{z})}{p(\mathbf{z},\mathbf{y})} \right] &= \mathbb{E}_q \left[\ln \frac{q_i(z_i)}{p(\mathbf{z},\mathbf{y})} \right] + \text{const} \\
&= \int q_i(z_i) \left\{ \ln q_i(z_i) - \prod_{j\neq i} \ln \exp q_j(z_j) \ln p(\mathbf{z}, y) dz_j \right\} dz_i + \text{const} \\
&= \int q_i(z_i) \ln \underbrace{\left[\frac{q_i(z_i)}{\prod_{j\neq i} \exp q_j(z_j) \ln p(\mathbf{z}, y) dz_j} \right]}_{\tilde{q}} dz_i.
\end{aligned} \tag{40}$$

Thus, $\mathbb{E}_q \left[\frac{q(\mathbf{z})}{p(\mathbf{z},\mathbf{y})} \right] = D(\mathbf{q}||\tilde{\mathbf{q}}) + \text{const}$ and optimized when $\tilde{q} = 1$. Therefore, we can obtain q_i^{opt} by denoting $\mathbb{E}_{\setminus i}$ as the average except z_i:

$$q_i^{\text{opt}}(z_i) = \frac{1}{Z_i} \exp\{\mathbb{E}_{\setminus i}[\ln p(\mathbf{z}, \mathbf{y})]\}, \tag{41}$$

this approximation scheme is called mean-field approximation as we seen in statistical mechanics.

3 Inference in complex network

3.1 Network Inference-missing link problem

Network analysis have been widely used in the different aspects, from protein-interactions to criminal determine social problem. However, reliabilities of network data is always a crucial concern, one of the most blatant example is the inaccuracy in protein interaction data. Inaccuracy of data will change the topology of the network, links may be missing or spurious. Some studies demonstrated that the inaccuracy of data will leads to a wrong estimation of network properties and therefore misleading conclusion. In this section, we will illustrate a method that we can use stochastic block model(SBM) to compute link reliability and use it to solve missing link problem.

Firstly, we denote the topology of an observed network as adjacency matrix $\mathbf{A}^0$[1], and n^0 is the number of observations(i.e. number of maximum edges might appear in the network) and n^1 is the number of edge observed. We also define the number of no-links pair as $n^\phi = n^0 - n^1$. From this, we can compute the probability of an edge exist between a pair of nodes (i, j) given $\mathbf{A}^0$ as:

$$p(a_{ij} = 1|\mathbf{A}^0) = \sum_{\mathcal{M}} \frac{p(\mathbf{A}^0|\mathcal{M})p(\mathcal{M})p(a_{ij} = 1|\mathcal{M}, \mathbf{A}^0)}{p(\mathbf{A}^0)}, \tag{42}$$

where $p(\mathbf{A}^0|\mathcal{M})$ is the likelihood that describe the probability of observed network $\mathbf{A}^0$ is given by model $\mathcal{M}$, $p(\mathcal{M})$ is prior corresponding to the probability of a model $\mathcal{M}$ being true and $p(a_{ij} = 1|\mathbf{A}^0)$ is the probability describe the reliability of a link. The entry of the adjacency can only be 0 and 1, therefore we can use Bernoulli process with a link exist probability p to simulate a given topology, where $\mathcal{M} := \mathbf{p}$. Use the likelihood function $p(\mathbf{A}^0|\mathbf{p})$ for a Bernoulli process and non-informative prior and we can derive the posterior $p(\mathbf{p}|\mathbf{A}^0)$ as:

$$p(\mathbf{p}|\mathbf{A}^0) = \frac{\mathbf{p}^{n^1}(1-\mathbf{p})^{n^\phi}}{\int_0^1 \mathbf{p}^{n^1}(1-\mathbf{p})^{n^\phi} d\mathbf{p}}. \tag{43}$$

Then the reliability of an edge $p(a_{ij} = 1|\mathbf{A}^0)$ can be computed from Equation 42 by using posterior and likelihood function(we might find using the beta-function is helpful while calculating integral):

$$p(a_{ij} = 1|\mathbf{A}^0) = \frac{\int_0^1 \mathbf{p} \times \mathbf{p}^{n^1}(1-\mathbf{p})^{n^\phi} d\mathbf{p}}{\int_0^1 \mathbf{p}^{n^1}(1-\mathbf{p})^{n^\phi} d\mathbf{p}} = \frac{n^1+1}{n^0+2} \approx \frac{n^1}{n} = p, \quad \text{in the large n}^0 \text{ limit.} \tag{44}$$

[1]Adjacency matrix store the information about the topology, we assume the network is symmetric and and each entry $A_{ij} \in \{0, 1\}$

3.1.1 Stochastic Block Model

In a stochastic block model, nodes are partitioned into different groups, the probability of two nodes been connected are only described by the probability of two groups been connected, i.e.

$$p(i \sim j|\text{SBM}) = p(a_{ij} = 1|\text{SBM}) = q_{\sigma_i \sigma_j}, \tag{45}$$

where σ_i are the membership of the group that node i belongs to. We can compute the likelihood function of an observed topology $p(\mathbf{A}^0|\sigma, Q)$ as follow:

$$p(\mathbf{A}^0|\sigma, Q) = \prod_{(i,j)\ a_{ij}=1} q_{\sigma_i \sigma_j} \prod_{(k,l)\ a_{kl}=0} (1 - q_{\sigma_k \sigma_l}) = \prod_{\alpha \leq \beta} q_{\alpha\beta}^{n^1_{\alpha\beta}} (1 - q_{\alpha\beta})^{n^{\phi}_{\alpha\beta}}. \tag{46}$$

Then we can compute the reliability $p(a_{ij} = 1|\mathbf{A}^0)$ by using Equation 43 as follows,

$$p(a_{ij} = 1|\mathbf{A}^0) = \frac{\sum_{\{\sigma\}} q_{\sigma_i \sigma_j} \int \prod_{\alpha \leq \beta} q_{\alpha\beta}^{n^1_{\alpha\beta}} (1 - q_{\alpha\beta})^{n^{\phi}_{\alpha\beta}} d\mathbf{Q}}{\sum_{\{\sigma\}} \int \prod_{\alpha \leq \beta} q_{\alpha\beta}^{n^1_{\alpha\beta}} (1 - q_{\alpha\beta})^{n^{\phi}_{\alpha\beta}} d\mathbf{Q}}, \tag{47}$$

where $d\mathbf{Q} = \prod_{\alpha \leq \beta} dq_{\alpha\beta}$ and $\{\sigma\}$ is all possible combinations that two groups may take. Consider the case $\alpha = \sigma_i, \beta = \sigma_j$, we can simplify the reliability into the following form:

$$p(a_{ij} = 1|\mathbf{A}^0) = \sum_{\{\sigma\}} \frac{n^1_{\sigma_i \sigma_j} + 1}{n_{\sigma_i \sigma_j} + 2} \prod_{\alpha \leq \beta} \frac{n^1_{\alpha\beta}!\ n^{\phi}_{\alpha\beta}!}{(n_{\alpha\beta} + 1)!}. \tag{48}$$

We can use statistical physics language to describe reliability:

$$p(a_{ij} = 1|\mathbf{A}^0) = \frac{1}{Z} \sum_{\{\sigma\}} \frac{n^1_{\sigma_i \sigma_j} + 1}{n_{\sigma_i \sigma_j} + 2} \exp(-H(\sigma)), \tag{49}$$

where $H(\sigma)$ is Hamiltonian given by configurations, i.e.

$$H(\sigma) = \sum_{\alpha \leq \beta} \left[\ln(n_{\alpha\beta} + 1) + \ln(\binom{n^0_{\alpha\beta}}{n^1_{\alpha\beta}}) \right]. \tag{50}$$

The partition function $Z = \sum_{\{\sigma\}} e^{-H(\sigma)}$ is not always exactly computable even for small networks, therefore we can use Metropolis algorithm to numerically compute partition function, the rate of convergence is low. At some extreme cases, we can approximate the reliability if a sharp peak exist as :

$$p(a_{ij} = 1|\mathbf{A}^0) \approx \frac{n^1_{\sigma^*_i \sigma^*_j + 1}}{n^0_{\sigma^*_i \sigma^*_j} + 2}, \tag{51}$$

where $\sigma^* = \text{argmax}\{p(\sigma|\mathbf{A}^0)\} = \text{argmax}\{H(\sigma)\}$.

3.1.2 Mixed-Membership Stochastic Blocks Model

Network inference method can be used to predict user preference with Mixed-Membership Stochastic Blocks Model. Similar to the previous approach, we assumes that the rating a group of users assign to an item is determined probabilistically by their group memberships, however, the users or the item may belong to a mixture of many different groups. Consider a rating problem, we use the bipartite network to simulate the rating process between users and items. We firstly define the links in the bipartite networks as $R = \{(u, m)\}$, if a link (u, i) appears it means a rating was given to an item m from user u. Furthermore, for each rating assigned $(u, m) \in R$, the rating is bounded between 1 to 5, i.e. $r_{um} \in S := \{1, 2, 3, 4, 5\}$, we will make sure that a rank will assign from user u to item m so that the probability is normalized i.e. $\sum_{r=1}^{5} p(r_{um} = r) = 1$. Our problem now simplifies to a missing-links dectection problem as discussed in the previous section. Given a set of obeserved links R^0, we need to determine the probability of a link appears between users u and m in different rating groups $r_{um} \in S$. To model the mix membership of the network, we need to define some more variables. Firstly, we define $\theta_{u\alpha}$ as the user u belong to

some intermediate groups β and the probability $\eta_{\beta m}$ of an item m belong to groups β and the probability $p_{\alpha\beta}(r_{um})$ as users groups α give item groups β a rank of $r_{um} \in S$. Since θ, η, p^r are probability so they are normalized i.e. $\sum_\alpha \theta_{u\alpha} = \sum_\beta \eta_{\beta m} = 1$. Then we can define write down likelihood function given by model as:

$$p(R^0|\theta, n, p) = \prod_{(u,m)\in R^0} \sum_{\alpha\beta} \theta_{u\alpha}\eta_{\beta m} p_{\alpha\beta}(r_{um}). \tag{52}$$

We need to infer the values of parameters η, θ and p that maximize the likelihood function. In the literature, Expectation-Maximization was the algorithm used to achieve the target. By introducing a latent variable $\omega_{um}(\alpha\beta)$ for implementing EM algorithm, $\omega_{um}(\alpha\beta)$ is an variational estimate of the probability that the rating r_{um} is from user u and item m belonging to the groups $\alpha\beta$ respectively. From Jensen's inequality $\log \mathbb{E}(x) \geq \mathbb{E}(\log x)$, we can rewrite log-likelihood function from Equation6 into the inequalities form:

$$\begin{aligned} \ln p(R^0|\theta, n, p) &= \sum_{(u,m)\in R^0} \ln \sum_{\alpha\beta} \frac{\theta_{u\alpha}\eta_{\beta m} p_{\alpha\beta}(r_{um})}{\omega_{um}(\alpha\beta)} \ \omega_{um}(\alpha\beta) \\ &\geq \sum_{(u,m)\in R^0} \sum_{\alpha\beta} \omega_{um}(\alpha\beta) \ln \frac{\theta_{u\alpha}\eta_{\beta m} p_{\alpha\beta}(r_{um})}{\omega_{um}(\alpha\beta)}. \end{aligned} \tag{53}$$

The lower bound hold with equality when (i.e. $\log \mathbb{E}(x) = \mathbb{E}(\log x)$) :

$$\omega_{um}(\alpha\beta) = \frac{\theta_{u\alpha}\eta_{\beta m} p_{\alpha\beta}(r_{um})}{\sum_{\alpha'\beta'} \theta_{u\alpha'}\eta_{\beta' m} p_{\alpha'\beta'}(r_{um})}. \tag{54}$$

The lower bound condition can be verified by inserting $\omega_{um}(\alpha\beta)$ from Equation 54 into Equation 53. By maximizing the log-likelihood function (53), with respect to η, θ and p and including Lagrange multipliers subject to the normalization constraints, we can obtain the update equation for $\theta_{u\alpha}$, i.e.,

$$\begin{aligned} \lambda &= \frac{\partial \ln p(R^0|\theta, n, p)}{\partial \theta_{u\alpha}} \\ &= \sum_{m\in\partial u} \sum_{\beta} \omega_{um}(\alpha\beta) \frac{1}{\theta_{u\alpha}}, \end{aligned} \tag{55}$$

Multiply $\theta_{u\alpha}$ both side and sum across all α both side we have:

$$\begin{aligned} \sum_\alpha \theta_{u\alpha}\lambda &= \sum_{m\in\partial u} \sum_{\alpha\beta} \omega_{um}(\alpha\beta) \\ \lambda &= \sum_{m\in\partial u} 1 = d_u \\ \theta_{u\alpha} &= \frac{\sum_{m\in\partial u} \sum_\beta \omega_{um}(\alpha\beta)}{d_u}, \end{aligned} \tag{56}$$

where $\partial u = \{m|(u,m) \in R^0\}$ denotes neighbors of user u, which is the movie that user u has ranked and d_u correspond to the degree of users(i.e. number of movie user ranked). By similar method used above and we can obtain η and p:

$$\begin{aligned} \eta_{\beta m} &= \frac{\sum_{u\in\partial m} \sum_\alpha \omega_{um}(\alpha\beta)}{\sum_{u\in\partial m} \sum_{\alpha\beta} \omega_{um}(\alpha\beta)} = \frac{\sum_{u\in\partial m} \sum_\alpha \omega_{um}(\alpha\beta)}{d_m} \\ p_{\alpha\beta}(r) &= \frac{\sum_{(u,m)\in R^0|r_{um}=r} \omega_{um}(\alpha\beta)}{\sum_{(u,m)\in R^0} \omega_{um}(\alpha\beta)}. \end{aligned} \tag{57}$$

From these Equations, we can perform EM-algorithm to calculate p, θ, η so that we can compute the likelihood function. For EM algorithm, we generally start from some initial estimates η^0, θ^0, p^0. At *(i) Expectation step*, we compute $\omega_{um}(\alpha\beta)$ for $(u,i) \in R$ from Equation54 and determine the maximum value for $\omega_{um}(\alpha\beta)$. (ii) *maximization step*, we use Equation 56-57 and ω from the *(i) Expectation step* to compute η, p, θ. Then repeat the above steps then finally a fixed point will be reached. For each EM step, the stochastic matrix is in linear in the size of the dataset. Typically the set of observations R^0 are sparse since only a small fraction of all possible user-item pairs have ratings, therefore, the algorithm can be used in the larger dataset.

3.1.3 Network inference result

To test the performance our model, we split the data into training set and test set. We trained our algorithm with the selected training set and test the performance of the algorithm with the test set. The links were removed from the original network and then use several trained different algorithms to make a predictions and so that the data can be compare. The performance of each algorithm was compared by their accuracy, mean absolute error(MAE), probability calibration with the true result and marginal calibration. Accuracy is defined by the fraction of the predicted rank which gives the true ranking from the data set. From the Figure (1a), it is obvious that the accuracy of MMSBM and SBM outperformed other traditional algorithms by their outstanding accuracy(from the left column) and also have a higher prediction precision(i.e. less fluctuations around mean, less mean square error). Probability calibration is a measure to suggest the confidence of the prediction compared to real probability[11], i.e. how frequently does the prediction the accurately capture the stochasticity of the data. Marginal distribution measures distance between the average probability for each ranking coincides with its actual probability obtain from the network. From the Figure 1 (b), we can see that MMSBM and SBM has a better calibrated probability than other algorithm.

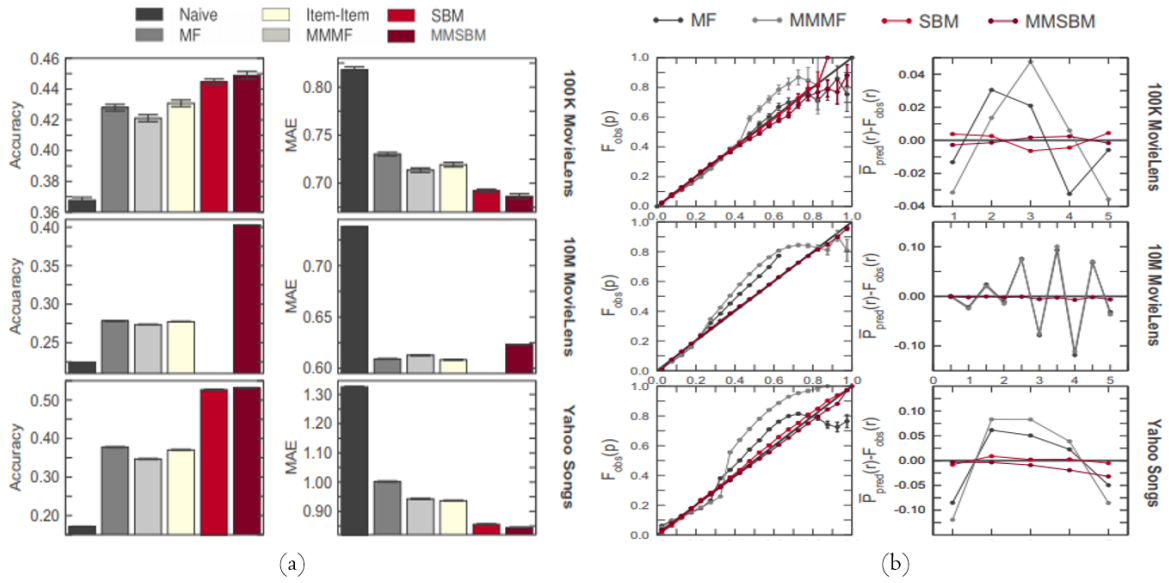

Figure 1: Accuracy of the algorithm(adapted from[8]) (a) The performance of different algorithms for predicting user–item ratings.The datasets involved are MovieLens 100K, Movielens 10M, Yahoo! Songs. The left column displayed the accuracy of the prediction(the fraction of ratings were equal to the prediction). The right column shows average mean absolute error between actual rating and the predicted outcome. The error bars in both column are corresponding to the standard error of the mean. (b) Left column, probability calibration for each model, where F^{obs} is the fraction of the predicted links and p is the predicted probability. The black solid line is the case that prediction exactly captured the stochasticity given by data. Right Column, marginal calibrating, the distance between the prediction against the fraction of the predicted links.

3.2 Model Selection

Model selection is a task to select most appropriate model from a given data set. In this case, we discuss some ways enable us to determine whether MMSBM or SBM is more appropriate :

1. *Option 1*: Compute likelihood ratio i.e.
$$\frac{P(\text{data}|\text{SBM})}{P(\text{data}|\text{MMSBM})}$$
2. *Option 2*: Compute posterior ratio i.e.
$$\frac{P(\text{SBM}|\text{data})}{P(\text{MMSBM}|\text{data})}$$

3. *Option 3* Compare the accuracy of the prediction given by algorithm with the data (See the last section)

In the large data set limit, the Option 1 and 2 will be identical since the effects from prior will vanished. However, increasing the parameter space will increase the likelihood function and cause over-fitting problem. Therefore, we introduce Bayesian information criterion to regularize the number of parameters,

$$BIC = k\ln(n) - 2\ln(\mathcal{L}). \tag{58}$$

where k is the number of the estimated parameters(the number of parameter space + 2 for estimating mean and variance), n is the number of the data points and $\mathcal{L}$ is the estimated maximum likelihood function. In general, A lower BIC value indicates a better fit [9].

4 High dimensional inference and Biological system

In an interactive model, our goals are to infer the correlations in the population density and to use machine learning/statistical learning to deduce the underlying mechanism, so predictive models can be produced for predicting future evolutions of the system. For instance, inferring effective connection for encoding/decoding problem for concerted activity of a neural population, inferring interactions between species from Lotka-Volterra equations and interactions in protein networks.

In general, biological network is more complicated than physics model. In physics model, we have uniform interactions(i.e. elements are identical to each other), so that the dimension of model(the number of parameters) will be low. The reproducibility of model are pretty decent and we can obtain good sampling, however, this is not case in biological model. In biological model, we usually will normally not have sufficient data compare to the number of parameters in the model and meanwhile, the experiment is costly, therefore, we need to introduce some techniques of high-dimensional inference . The goal of high-dimensional inference is that the compression of data, i.e. how to eliminate the indirected correlations from the data and have a sparser representation and use them to construct the generative model that agrees with experiments. Classically we know that if the parameter numbers is fixed, if we perform asymptotic inference(it is called asymptotic means that if the number of data size go to infinity), then we know the the error in our parameter estimation will decreased as $O(N^{-1/2})$, but the problem arose since the number of parameter is usually not fixed, which means, the size of parameter space will grow with the size of dataset. To solve this problem, we use high-dimensional inference. Let us consider the following example as an illustration. Assume we have n samples of data with p variables as $\mathbf{d}^{(s)} = \left(d_1^{(s)} \cdots d_p^{(s)}\right)$ and each observation d is i.i.d. and follows an Gaussian distribution as followings:

$$p(\mathbf{d}^{(s)}|\tau) = \frac{\sqrt{\det(\tau)}}{2\pi^{\frac{p}{2}}} \exp\left[-\frac{1}{2}\sum_{ij} d_i^{(s)}\tau_{ij}d_j^{(s)}\right]. \tag{59}$$

Therefore, like traditional Maximum Likelihood Estimation method :

$$\ln p(\mathbf{d}|\tau) = \frac{1}{n}\sum_{s=1\cdots n}\ln p(\mathbf{d}^{(s)}|\tau) = \frac{1}{2}\ln\det(\tau) - \frac{1}{2}\sum_{ij}\tau_{ij}\frac{1}{n}\sum_s d_i^{(s)}d_j^{(s)}, \tag{60}$$

where the correlation can be easily computed from the data and easy to perform MLE $= -1/2n\ \sum_s d_i^{(s)}d_j^{(s)} + 1/2\ (\tau^{-1})_{ij}$. The MLE method is good but it has two problems, firstly, if the likelihood function is a non-decreasing function then there is no local maximum. Secondly, if there are some corruptions in sampling noise, then the covariance is no longer reliable, (unreliable inversion). For sufficiently large data set n, the estimate of correlations converges to true estimation, according to CLT, the order of error on the estimated correlation is scaling with $O(1/\sqrt{n})$ errors in inverting matrix is of the order $(p/n)^{1/2}$, which means if number of parameter p is increasing as n, the error will not converge so we have bad estimation. Instead of using Gaussian likelihood, we can use the Ising-like distribution with local field, i.e.:

$$p(\mathbf{d}|\mathbf{J},\mathbf{h}) = \frac{1}{Z(\mathbf{J},\mathbf{h})}\exp\left[-\frac{1}{2}\sum_{ij} d_i J_i j d_j + \sum_i h_i d_i\right], \qquad d_{ij} \in \{0,1\}. \tag{61}$$

We use Boltzmann learning to study the distribution. Suppose we have some initial estimation of interaction **J** and field estimates, we can compute the expected spin:

$$\langle d_i \rangle = \sum_{\mathbf{d}} \frac{d_i}{Z(J)} \exp\left[\sum_{i<j} J_{ij} d_i d_j + \sum_i h_i d_i\right]. \tag{62}$$

The partition function is generally hard to calculate, since **d** has 2^p (p in number of spin) configurations, instead, we can use mean-field approximation (as mentioning in the first few section) or Monte-Carlo simulation. Once we have obtained the partition function, we can recalculate Hamiltonian and update our model with the correlations obtained from data accordingly:

$$J_{ij} \to J_{ij} - a(\langle d_i d_j \rangle - c_{ij}), \qquad h_i \to h_i a'(\langle d_i \rangle - m_i), \tag{63}$$

where a, a' is the learning constant. The likelihood will converge to the true one if it is a concave function, however, it have several drawbacks, it is very slow in evaluating the partition function via the Monte-Carlo and rely-on the result from Monte-Carlo simulation. Furthermore, if the mode of the likelihood function is flat, again the above method is not the best, we therefore introduce pseudo-likelihood algorithm (see paper *neighborhood selection, 2006*). Again our aim is to infer the true **J** and **h** from the data. Now we only consider one site, i and its interaction to neighborhood, and we can write the conditional probability as:

$$p^{(i)}(d_i | d_{j\neq i}) = \frac{e^{H^{(i)}(d_{j\neq i})}}{1 + e^{H^{(i)}(d_{j\neq i})}}, \tag{64}$$

where $H^{(i)}(d_{j\neq i}) = h_i + \sum_{j\neq i} J_{ij} d_j$ is the effective Hamiltonian(similar to the first chapter), therefore the pseudo-likelihood can be written as:

$$p_l(\{d_i\}|J, h) = \prod_i^n p^{(i)}(d_i | \{d_{j\neq i}\}). \tag{65}$$

The method is much simpler since it does not involved computing the partition function. Notice, the inferred J_{ij} is not symmetrical since the site will have different neighbors. Again, we can evaluative the $\ln p_l$ with respect to h_i from what we find from each data set s:

$$\begin{aligned} \frac{\partial \ln p_l}{\partial h_i} &= \frac{1}{n} \sum_s \left(d_i^{(s)} - \frac{e^{H_i^{(s)}}}{1 + e^{H_i^{(s)}}} \right) = \langle d_i \rangle_{\text{data}} - \ \langle d_i \rangle_{\text{model}} \\ \frac{\partial \ln p_l}{\partial J_i j} &= \langle d_i d_j \rangle_{\text{data}} - \ \langle d_i d_j \rangle_{\text{model}} \end{aligned} \tag{66}$$

This is for the case for $d_i \in \{0, 1\}$, where the spin can be $s_i \in \{-1, 1\}$, we can compute the mean by using the mean-field approximation. However, we need to regularize the dimension of J, as it may have infinite values. To achieve this we will introduce a cost function $\mathcal{C}(h_i, \{J_{ij}(i \neq j)\})$, which is related to our prior knowledge:

$$\mathcal{C}(h_i, \{J_{ij}(i \neq j)\}) = -\frac{1}{n} \sum_{\{i\}} \ln p^{(i)}(d_i | \{d_{i\neq j}\}) + r_1 \sum_j |J_{ij}| + r_2 \sum_j J_{ij}^2 \tag{67}$$

The first summation involved r_1 is called L1 penalty, it will set the small J with zero, it kills noise and the fake correlation will disappear, therefore, regularize network to be sparse and the second sum is called L2 penalty which regularize the extreme value, if the data set is large so we have more evident to ensure the correlation is true, therefore we set r_2 close to zero, but in a smaller dataset a high correlation is more likely to be obtained and it not necessary been statistically correct we therefore will penalize it to prevent it gives wrong likelihood function.

4.1 Other Unsupervised machine learning method

4.1.1 Principal Component Analysis

Principal component analysis can be used to obtain whether the interactions occur in the network or to detect correlations in a sequence of multi-dimensional data. Consider a vector of standard Gaussian normal variable with zero mean and unit variance, each component is independent to each other with the covariance matrix $C = \tau$, where $\tau = \mathrm{I}d$ is an identity matrix. The resulting distribution is isotropic(see Figure 2 (a)). Now if we assume

there is a larger variance in a specific direction, $|\mathbf{e}\rangle$ in the N-dimensional space(shortly speaking, there are some correlations exist between some variables)i.e. $\tau = \mathsf{I} + s/(1+s)|\mathbf{e}\rangle\langle\mathbf{e}|$ see Figure 2(b)), where $s > 0$. We call $|\mathbf{e}\rangle$ principal component and it can interpreted as *collective mode of variation* of the data. If we denote the random variable in the vector form of $\sigma_{\mathbf{e}} = \sum_{i=1}^{N} \sigma_i \mathbf{e}_i$, the variance be calculated from $V_e = \langle\mathbf{e}|\mathsf{C}|\mathbf{e}\rangle = 1 + s$, since if $s > 0$. We know for an independent variable, the variance $V_{e_i} = 1$. If there is some special dimension correlations, we know the variance of this direction is greater than one, therefore, our question simplifies to find the maximum eigenvalue and the associate top eigenvector as followings,

$$\max \langle\mathbf{e}|\mathsf{C}|\mathbf{e}\rangle\,, \qquad \text{with } \langle\mathbf{e}|\mathbf{e}\rangle = 1. \tag{68}$$

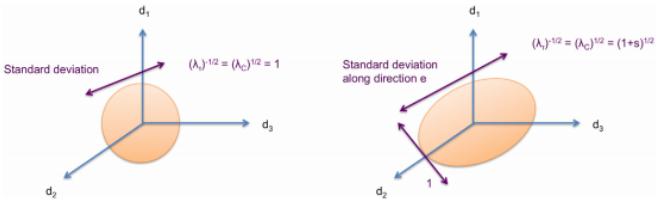

Figure 2: Probability density (adapted from [12])contours for (left) the null model ($\tau = \mathsf{I}d$) and (right) the principal-component model ($\tau = \mathsf{I} + s/(1+s)|\mathbf{e}\rangle\langle\mathbf{e}|$)

The top eigenvector can be inferred from maximum likelihood estimation defined at Equation 6 with the empirical matrix $\hat{\mathsf{C}}$ computed from the data:

$$p(\mathbf{d}^{(s)}|\tau) = \frac{\sqrt{\det(\tau)}}{2\pi^{\frac{p}{2}}} \exp\left[\frac{s}{2(1+s)} \sum_{ij} \mathbf{e}_i \frac{1}{n} \left(\sum_s d_i^{(s)} d_j^{(s)}\right) \mathbf{e}_j\right]. \tag{69}$$

If we maximize the likelihood, we can estimate the the top component, however, the top component from the empirical correlation agrees with the true top components only if the sampling size is infinity. This can be demonstrated easily if we consider a sequence of independent random variables(null models). If the correlation matrix is an identity matrix, there is an unique eigenvalue for this correlation matrix. However, if we use the empirical correlations obtained from data, we will obtain a different eigenvalues from each observation therefore, spectral density of eigenvalues. We can derive the spectral density in many ways(one of derivations see CS03), here we just state the results, in the limit that the number of samples, n tends to infinity and the number of variables p tend to infinity, the spectral density will depend on the ratio $r = p/n$:

$$\rho(\lambda) = \begin{cases} \frac{\sqrt{(\lambda_+ - \lambda)(\lambda - \lambda_-)}}{2\pi r\lambda} & \text{for } r < 1, \\ \frac{\sqrt{(\lambda_+ - \lambda)(\lambda - \lambda_-)}}{2\pi r\lambda} + (\frac{1}{r} - 1)\delta(\lambda) & \text{for } r \geq 1. \end{cases} \tag{70}$$

where $\lambda_\pm = (1 \pm \sqrt{r})^2$ back to the special direction correlated problem. In the weak noise condition $r < s^2$, we can obtain the largest eigenvalue as the peak . Oppositely, in the strong noise condition, $r > s^2$ the peak is no longer distinguishable and the critical noise level is $r = s^2$(can be think as a phase transition problem), which is the same for more eigenvalues greater than 1 (i.e. more peaks). In order to deal with the noise level $r > s^2$, we can introduce prior information to infer the top component:

$$p(V(\mathbf{e}_i)) = \exp\left(-\sum_i V(\mathbf{e}_i)\right), \tag{71}$$

the prior information(prior potential over all principal component) can help us to beat the noise level, if we use the non-negative prior, then the boundary can be extended to $r < 2s^2$. We can also use other prior such as sparse and large entry prior to beat the boundary. However, the difficulties is that the strength of prior we need to chose, if inappropriate prior was chosen, the critical threshold r_c will be distorted.

4.1.2 Auto-encoder

PCA is powerful method as long as the data is normally correlated, however, for non-Gaussian correlation, we need to apply other dimensional reduction method, such as auto-encoder. The procedure of auto-encoder can be split

into two steps, the first step is called encoding, which compress the input into a latent-space representation and then decode, reconstruct the input from the latent space representation as output(See Figure 3(b)). We can represent the transformation as follow:

$$\mathbf{V} \Rightarrow \mathbf{h} = F(\mathsf{M} \cdot \mathbf{V}) \Rightarrow \mathbf{V}' = G(\mathsf{M}' \cdot \mathbf{h}), \tag{72}$$

where M is a matrix keep the dimension of the input or output and F, G is the functional transformation. We need to minimize the cost function D to ensure the good quality of output as:

$$D = \sum_s |\mathbf{V}^{(s)} - G(\mathsf{M}' \cdot F(\mathsf{M} \cdot \mathbf{V}^{(s)}))|^2. \tag{73}$$

One of applications of auto-encoder is removing noise in the noisy data input. We can improve the quality of the output by increasing the size of the pools of hidden-space representations and enforce the sparsity. Loosely speaking, enforce the sparsity means that the the hidden layer representation will have less interference between each other, however, the efficiency of the algorithm will be much lower. We can also introduce another application of sparse coding, sparse dictionary learning in image de-noising. The idea is similar to each neuron in brain is only to recognize certain edge. We decompose our input picture into pixels and then train a hidden layer of sparse feature-space representation and then use it to reconstruct the output that minimize the cost function (73) over M, M'. In this case, we can use linear function which $G(x) = x, F(x) = x$, which means the output is going to a linear combination of the hidden-space representation. Let ϕ_i represent elements i in feature space (or an atom in the machine learning language) and output of a pixel in a image can be written as: $V^{(s)} = (\sum_i w_i \phi_i)/\sqrt{\sum_i w_i^2}$(sparse here means many of elements in feature space are not used, i.e.$w = 0$).

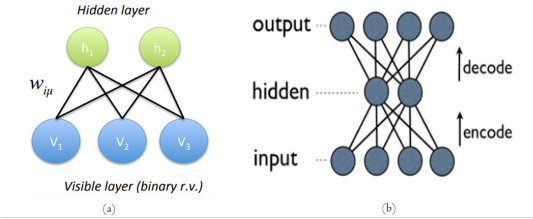

Figure 3: Illustration of (a) Restricted Boltzmann Machine and (b) Autoencoder. Adapted from [12]

4.1.3 Restricted-Boltzmann machine

Restricted Boltzmann machine is a graphical model that commonly used in dimension reduction problem. RBM construct a bipartite types neural network, where nodes are only connected to nodes from different groups, this is the *restriction* meaning. There are two layers in RBM, one is visible layer and another one is hidden layer. The nodes in visible layer will only give binary random variables $v_i \in \{0, 1\}$(In image processing problem, 0 can represent black color and 1 represent white color) and the strength of the interaction between these two nodes are described by wight matrix w_{ui}. The ultimate goal is to train the hidden layer from visible layer and predict patterns if a partial input is given. We assume the joint distribution $P(\mathbf{h}, \mathbf{v})$ is Boltzmann distributed according to Energy function given in the Hopfield model and our goal is to obtain the marginal distribution $P(\mathbf{v})$:

$$P(\mathbf{h}, \mathbf{v}) = \frac{1}{Z} \exp\left[-E((\mathbf{v}, \mathbf{h}))\right], \qquad \text{where } E((\mathbf{v}, \mathbf{h})) = -\sum_i g_i v_i + \sum_\mu U_\mu(h_\mu) - \sum_{i\mu} w_{i\mu} v_i h_u. \tag{74}$$

We choose $U_\mu(h_\mu)$ as a quadratic potential and we can rewrite the marginal distribution as:

$$P(\mathbf{v}) \int \prod_\mu dh_\mu P(\mathbf{v}, \{h_\mu\}) = \frac{1}{Z_{\text{eff}}} \exp\left[-E_{\text{eff}}(\mathbf{v})\right], \qquad \text{where } E_{\text{eff}} = -\sum_i g_i v_i + \frac{1}{2} \sum_{i,j} \left(\sum_\mu w_{i\mu} w_{j\mu} \right) v_i v_j. \tag{75}$$

The Boltzmann machine is therefore similar to Hopfield model if the weight matrix w_u as one pattern ξ_u. It is noticeable if the potential is non-quadratic, then the effective Hamiltonian is calculated by multi-body interactions(instead of two-body interaction). We can therefore compute the hidden unit inputs I_μ^H, which can be used to infer to the hidden unit as following:

$$I_\mu^H = \sum_i w_{i\mu} v_i, \qquad P(h_\mu | I_\mu^H) \propto \exp^{\left[-U(h_\mu) + I_\mu^H h_\mu\right]}. \tag{76}$$

We maximize the likelihood function in Equation (76) with respect to hidden unit h, where we can obtain that $dU(h)/dh = I^H$. In the quadratic potential case, we have linear hidden unit, i.e. $h = I^H$. There are other choices of hidden units depend on the potential such as Rectifier Linear unit(ReLU) and Bernoulli unit(For ReLU, $h = \max\{I^H - y, 0\}$, where y is a constant. For Bernoulli, $h = \theta(I^H - \theta_b)$), which is a heavy step function depend on θ_b. We want to maximize the log-likelihood function $\ln p(\mathbf{v}|\theta)$ over the parameter space so we use pseudo-likelihood method for Boltzmann learning algorithm, as discussed previous section by recognizing $g_i \to h_i, w_{iu} \to J_{ij}$ with a data set $V = \{v_i^b, i = 1, \cdots, N, b = 1 \cdots B\}$. To compare accuracy of different choices of between the units, we train the machine with MNIST dataset with $60,000$ image of digits of size 28×28, for both machine we have 400 number of units and we find ReLU potential have a higher log-likelihood value than Linear RBM, which is an evidence of ReLU is more appropriate model according to model selection rules. Meanwhile, the digits inferred from ReLU is less noisy and MCMC are mixing much more faster. The reason for it is ReLU will have a more sparser hidden layer since it will filter out the low value signal so that less interference between the dataset we will obtain.

5 Summary

In this lecture notes, we have discussed several methods and applications of inference method, from using Bayesian inference to compute regression problem to approximate inference to deal with intractable probability distribution. The technique of how to use stochastic block models in network to predict missing link was illustrated and the inference result shows the family of SBM outperform all other algorithm. In the end section, we illustrate the techniques used in the high-dimensional inference, from PCA applied to Gaussian correlated data to Auto-encoder to non-Gaussian correlated data and some insights of dictionary learning. We have also discussed the mechanism behind restricted Boltzmann machine and how it was related to Hopfield model as a dimension reduction technique with its application in recognizing digits. Overall, inference is an advanced topics that has wide applications in many topic with impressive accuracy, which is a field have a great potential.

References

[1] Balding D., *Inference in complex systems*, Interface Focus. 2011;1(6):805-806. doi:10.1098/rsfs.2011.0074.

[2] Nasrabadi, N. M., *Pattern recognition and machine learning*, Journal of electronic imaging, 16(4), 049901.

[3] Sollich P. , Lecture notes of *Element Statistical Learning*, Kings College London

[4] C. E. Rasmussen, C. K. I. Williams, *Gaussian Processes for Machine Learning* The MIT Press, 2006. ISBN 0-262-18253-X.

[5] José M., *Gaussian Processes for regression: a tutorial*,

[6] M. Opper, *An estimator for the relative entropy rate of path measures for stochastic differential equations*, 2016 Journal of Computational Physics

[7] Besag.J , *Statistical Analysis of Non-Lattice Data", The Statistician*, 24 (3): 179–195, JSTOR 2987782

[8] Sales-Pardo M., *Accurate and scalable social recommendation using mixed-membership stochastic block models*

[9] *Chapter 32:Emerging Business Intelligence Framework for a Clinical Laboratory Through Big Data Analytics*, Emerging Trends in Computational Biology, Bioinformatics, and Systems Biology, Hamid Arabnia Quoc Nam Tran, ISBN: 9780128026465

[10] Jaynes, E. T. , *Prior Probabilities* . IEEE Transactions on Systems Science and Cybernetics. 4 (3): 227–241. doi:10.1109/TSSC.1968.300117.

[11] A Niculescu-Mizi, *Obtaining Calibrated Probabilities from Boosting*, Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI2005)

[12] R. Monasson, *Statistical physics and representations in real and artificial neural networks*

6 Critical analysis - Multilayer Stochastic Block models Reveal the Multilayer Structure of Complex Network

Stochastic Block model(SBM) is one of common classes of probabilistic generative network model that can be used to explain how the group structure was formed in the real complex network. In the SBM, nodes are assigned to some group ships and the probability of two nodes being connected is only dependent on their membership. Furthermore, the probabilistic form of SBM provides a method to detect missing link/spurious links from the empirical network data, for instance, to determine unknown interaction between drugs. However, single-layer SBM has limitations according to the assumptions of that the interactions in the network is only controlled by single mechanism, whereas, the real networks are generally a projection of multi-layer networks. As many researches pointed out, the existence of multilayer layer interactions will heavily impact the dynamical processes occurred at the network, therefore, ignorance of the structure will provide a wrong inference result. The interactions occur in multilayer is generally not measurable, though the projection of the interaction can be observed. In this paper, a new family of multi-layer SBM model is designed to extend generative model to multi-layer, with the tools borrowed from statistical physics.

The paper consider two types of two-layer interaction mechanism: *(i)* the AND combination of two layers, where the nodes i and j at the aggregated network are connected if and only if they are connected at the both layers *ii* the OR combination, where the node i and j at the aggregated network are connected if at least in one layer these two nodes are connected. In this review only AND mechanism are discussed. The primary goal is to identify the pair of partition $(\mathcal{P}_1, \mathcal{P}_2)$ from layer 1 and layer 2 that best describe observed aggregated topology, with no prior information about the layers, i.e. maximize the the posterior function $P(\mathcal{P}_1, \mathcal{P}_2|A^O)$, where A^O is the observed aggravated network. By using an uniform non-informative prior, $P(\mathcal{Q}_1, Q_2, \mathcal{P}_1, \mathcal{P}_2)$, where $\mathcal{Q}$ is a matrix describe the probability of interaction occur in the same layer between two different groups. The posterior of AND mechanism for two layer partitions $P(\mathcal{P}_1, \mathcal{P}_2|A^O)$ are given by:

$$P_{\text{AND}}(\mathcal{P}_1, \mathcal{P}_2|A^O) \propto \sum_{\{m_{rs}\}} \frac{\prod_{r,s}(-1)^{m_{rs}}\binom{n^0_{rs}}{m_{rs}}}{\prod_r(n^1_r + m_r + 1)\prod_s(n^1_s + m_s + 1)} \tag{77}$$

where $m_{rs} = 0, \cdots, n^0_{rs}$ is the discrete possible values, $r = \alpha\beta$ ($\alpha\beta$ is the membership of a pair of nodes in the first layer, $\gamma\delta$ is for the second layer) and $s = \gamma\delta$, $m_r = \sum_s m_{rs}$ and n^1_r is the number of links between pairs of nodes r in layer 1 and n^0_r is the number of no links between pairs of nodes r in layer 1 and pairs of nodes s in layer 2. Therefore, if we can maximize Equation 77, we can find the optimal partition $\mathcal{P}_1, \mathcal{P}_2$ so that entangled observed aggregated network into two layers. Furthermore, Equation 77 enable us to a method to compare the multi-layer SBMs with regular SBM by comparing posterior ratio, however, the above equation is numerically intractable, even the total numbers of group is not large, hence, approximation is needed for computing the quantity.

To understand how approximation works, we need to know that multilayer SBMs can be represented as a single layer SBM. We call the single-layer representation as the intersection partition $\mathcal{P}_I$. The approximation used is that assuming the probability of group-to-group connections between different pairs of groups in $\mathcal{P}_I$ are independent. The approximated representation will no longer able to give us an unique optimal multi-layer partitions from the observed network, instead, it allows us to compare the predictive accuracies between the single-layer SBM and the multi-layer SBM in the problem of detecting the missing/spurious links in a noisy set of data from computing reliability of a link as $R_{ij} = P(A_{ij} = 1|A^O)$ for any set of $\mathcal{M}$ models from the following expression:

$$R^{\mathcal{M}}_{ij} = \frac{\int_{\mathcal{M}} dM P(A_{ij} = 1|M)P(A^0|M)P(M)}{Z}, \tag{78}$$

where Z is the normalization constant. With the approximation mentioned above, the reliability of links in AND model can therefore be calculated by:

$$R^{\text{AND}}_{ij} = \frac{1}{Z}\sum_{\mathcal{P}_I}\left(\frac{n^1_{\sigma_i\sigma_j} + 1}{n_{\sigma_i\sigma_j} + 2}\frac{\sum_{k=n^1_{\sigma_i\sigma_j}+2}^{n_{\sigma_i\sigma_j}+2}\frac{1}{k}}{\sum_{k=n^1_{\sigma_i\sigma_j}+1}^{n_{\sigma_i\sigma_j}+1}\frac{1}{k}}D(\mathcal{P}_I)e^{-\mathcal{H}(\mathcal{P}_\mathcal{I})}\right) \tag{79}$$

where the sum is over all possible intersection partition, $n^1_{\alpha\beta}$ is the number of links between the intersection partition, $n_{\alpha\beta} = n^0_{\alpha\beta} + n^1_{\alpha\beta}$ is the number of possible links between pairs of nodes in groups α and β and $D(\mathcal{P}_\mathcal{I})$ is the number of pairs $(\mathcal{P}_1, \mathcal{P}_2)$ that have the same intersection partitions(i.e. multiplicity/the degeneracy of partition). The

energy function $\mathcal{H}(\mathcal{P}_I)$ is computed by:

$$\mathcal{H}(\mathcal{P}_I) = \sum_{\alpha\leq\beta}\left[\ln(n_{\alpha\beta}+1) + \ln\binom{n_{\alpha\beta}}{n^1_{\alpha\beta}} - \ln\left(\sum_{k=n^1_{\alpha\beta}+1}^{n_{\alpha\beta}+1}\frac{1}{k}\right)\right]. \tag{80}$$

The link reliability of AND model is similar to the ensemble average of an observable in statistical physics, which can be numerically computed by Markov-Chain Monte-Carlo simulation[2].

To verify if the approximated two-layer SBM can indeed detect if the observed network is made by multi-layer model, we construct a set of two-layer test networks ensembles that have well defined block structure in both layers from the AND an OR model. The network ensemble was only characterized by two variables i the low-to-high connectivity ratio and the average connectivity of nodes. For testing the predictive power for missing link problem, a fraction f of the links was removed and compute the fraction of times that a removed link has a higher reliability(False negative rate) than a link that originally is not in the network(True negative rate) but for spurious link we add the f fraction of links and then we compute fraction of the links has the lower reliability (False positive rate) than a link is not presented(True positive rate). The reliability for AND model can be computed from Equation (79). From the simulation, it has demonstrated if the network is working as a two-layer process, the prediction given by two-layer prediction out-performed the single layer approach, especially in the case when the number of distinct node groups in the intersection partition and the connectivity is high and the noise level f are moderate or low. The test FROM synthetic network illustrated the approximated algorithm is indeed provide us a better model under. Therefore, the method was applied to study a real network to suggest whether the multi-layer aggregation mechanism better describe the observed topology. Original paper applied their method to study eight different real world networks but only two of them are listed in the summary(see caption of Figure 2). The result in Figure (4) suggested that two layer AND models are more suitable to describe the real network, based on their accuracies in detecting missing/spurious links were consistently higher than single layer stochastic block model, especially in the low noise limit (small f).

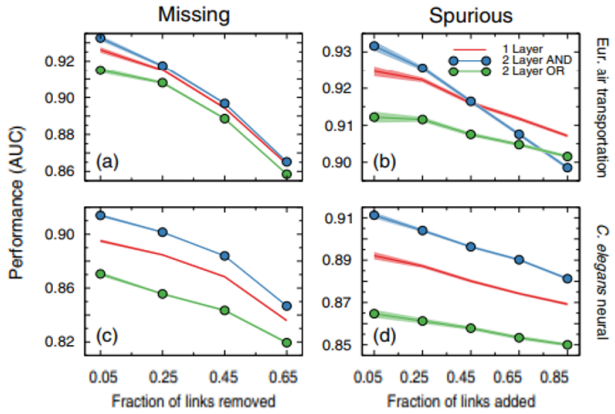

Figure 4: Performance of identification of missing links(a),(c) and spurious links (b),(d)(adapted from the paper) via using AUC on the network of the air transportation network in Eastern Europe and neural network of Caenorhabditis elegans.

To summarize, the project introduce a probabilistic multi-layer SBM family that generalized the single-layer SBM, in the case that the interactions(links) in the network are an aggregation of different mechanisms. Since the probabilistic solution is intractable even for two-layer aggregation mechanism for a small network, an approximation was made so we can objectively determine whether the observed network is an aggregation of multi-layer network. The result in the paper suggests the real network is indeed an multi-layer network, despite there are some limitations left, firstly, the approximation lost the detailed information where the topology of network can no longer to uniquely determined. Meanwhile, the author also figured out that even with the approximation scheme was used, the algorithm is computationally expensive and not well suited to deal with the large network. However, author are confident that in the future the structure of the network will be better understand.

[2] AUC is an indicator of how good prediction is in binary classification problem, if it is close to one then it means the classification quality is quite high