Exact learning curves for Gaussian process regression on community random graphs

Matthew J. Urry
Department of Mathematics
King's College London
London, WC2R 2LS, U.K.
matthew.urry@kcl.ac.uk

Peter Sollich
Department of Mathematics
King's College London
London, WC2R 2LS, U.K.
peter.sollich@kcl.ac.uk

Abstract

We study learning curves for Gaussian process regression which characterise performance in terms of the Bayes error averaged over datasets of a given size. Whilst learning curves are in general very difficult to calculate we show that for discrete input domains, where similarity between input points is characterized in terms nodes on a graph, accurate predictions can be obtained. These should in fact become exact for large graphs drawn from appropriate random graph ensembles. We focus on two types of ensemble. One is obtained by specifying (arbitrarily) the degree distribution and leads to sparse graphs, where each node is connected only to a finite number of others. The other is a community graph ensemble where we assume communities joined by a similar sparse superstructure. The calculation of the learning curves is based on translating the appropriate belief propagation equations to the graph ensemble. We demonstrate the accuracy of the predictions for Poisson (Erdos-Renyi) graphs and give some numerical results showing the need for a community orientated derivation of the learning curve.

1 Introduction

Learning curves are a convenient way of characterising the performance that can be achieved with machine learning algorithms: they give the generalisation error ϵ as a function of the number of training examples N, averaged over all datasets of size N under appropriate assumptions about the data-generating process. Such a characterization is particularly useful in the case of non-parametric approaches such as Gaussian processes (GPs) [1], where in contrast to the parametric case [2] there is no generic classification of possible learning curves.

Here we study GP regression, where a real-valued output function $f(x)$ is to be learned. Qualitatively, GP learning curves are relatively well understood for the scenario where the inputs x come from a continuous space, typically $\mathbb{R}^n$ [3, 4, 5, 6, 7, 8, 9]. However, except in the limit of large N, or for very specific situations like one-dimensional inputs [3], the learning curves cannot be calculated exactly. Here we show that this *is* possible for discrete input spaces where similarity between input points can be represented as a graph whose edges connect similar points, inspired by work at last year's NIPS that developed simple approximations for this scenario [10].

In section 2 we give a brief overview of GP regression and summarize the approximation for the learning curves used in previous work [11, 10, 4]. Section 3 then explains our method: following a similar approach in [12] for random matrix spectra, we write down the belief propagation equations for a given graph in the form normally used in the cavity method [13] of statistical mechanics, and then translate them to sparse graphs drawn from a random graph ensemble. Section 4 generalises this derivation to community random graphs of the type seen in [14, 15]. Because for sparse random graphs typical loop lengths grow with the graph size, the belief propagation equations and hence our

learning curve predictions should become exact for large sparse graphs; the same will hold for the community graphs given that we account exactly for the short loops inside each community.

Section 5 compares the cavity predictions with simulation results for Poisson (Erdos-Renyi) graphs. The new predictions are indeed very accurate, and substantially more so than previous approximations. Section 5 then provides numerical results to demonstrate the need for the more general community random graph predictions given in section 4. We compare the generalisation error of a community ensemble to two related sparse graph ensembles that are obtained by fixing appropriate degree distributions. Finally, section 6 summarises our results and discusses open questions and directions for future work.

2 GP regression and approximate learning curves

A GP is a Gaussian prior over functions f with a fixed covariance function (kernel) $\boldsymbol{C}$ and mean function (assumed to be $\mathbf{0}$)[1]. In the simplest case the likelihood is also Gaussian, i.e. we assume that the outputs y_μ in a set of examples $D = \{(i_1, y_1, \ldots, (i_N, y_N)\}$ are obtained by corrupting the clean function values f_{i_μ} with i.i.d. Gaussian noise of variance σ^2. Then the posterior distribution over functions is, from Bayes' theorem $P(f|D) \propto P(f)P(D|f)$:

$$P(f|D) \propto \exp(-\frac{1}{2}\boldsymbol{f}^T\boldsymbol{C}^{-1}\boldsymbol{f} - \frac{1}{2\sigma^2}\sum_{\mu=1}^{N}(y_\mu - f_{i_\mu})^2) \tag{1}$$

We consider GPs in discrete spaces, where each input is a node of a graph and can therefore be given a discrete label i as anticipated above; f_i is the associated function value. If the graph has V nodes, the covariance function is then just a $V \times V$ matrix.

A number of possible forms for covariance functions on graphs have been proposed. We will focus on the relatively flexible random walk covariance function [16],

$$\boldsymbol{C} = \frac{1}{\kappa}((1 - a^{-1})\boldsymbol{I} + a^{-1}\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2})^p \qquad a \geq 2, \quad p \geq 0 \tag{2}$$

Here $\boldsymbol{A}$ is the adjacency matrix of the graph, with $A_{ij} = 1$ if nodes i and j are connected by an edge, and 0 otherwise; $\boldsymbol{D} = \text{diag}\{d_1, \ldots, d_V\}$ is a diagonal matrix containing the degrees of the nodes in the graph ($d_i = \sum_j A_{ij}$).The constant κ will be chosen throughout to normalise $\boldsymbol{C}$ so that $\frac{1}{V}\sum_i C_{ii} = 1$, which corresponds to setting the average prior variance of the function values to unity.

Our main concern in this paper are GP learning curves in discrete input spaces. The learning curve describes how the average generalisation error (mean square error) ϵ decreases with the number of examples N. Qualitatively, it gives the rate at which one would expect a GP to learn a function in the *average case*. The generalisation error on an ensemble of graphs is given by

$$\epsilon = \langle \frac{1}{V}\sum_i (\bar{f}_i - f_i)^2 \rangle_{f|D,D,\text{graphs}} \tag{3}$$

where $\boldsymbol{f}$ is the uncorrupted (clean) teacher or target function, and $\bar{\boldsymbol{f}}$ is the posterior mean function of the GP which gives the function values we predict on the basis of the data D. It is worth noting that the generalisation error for a graph ensemble contains an additional average over this ensemble. As is standard in the study of learning curves we have assumed a matched scenario where the posterior $P(f|D)$ for our predictions is also the posterior over the underlying target functions. The generalisation error is then the Bayes error, and is given by the average posterior variance.

Sollich [4] and later Opper [7] with a more general replica approach showed that for continuous input spaces a reasonable approximation to the learning curve could be expressed as the solution of the following self-consistent equation:

$$\epsilon = g\left(\frac{N}{\epsilon + \sigma^2}\right), \qquad g(h) = \sum_{\alpha=1}^{V}(\lambda_\alpha^{-1} + h)^{-1} \tag{4}$$

[1]We focus on the zero prior mean case throughout. All results translate fairly straightforwardly to the non-zero mean case, but this complicates the algebra without leading to substantially new insights.

Here the λ_α are appropriately defined eigenvalues of the covariance function. The motivation for our study is work presented at NIPS2009 [10], which demonstrated that this approximation can also be used in discrete domains, but is not always accurate. Studying random walk and diffusion kernels [16] on random regular graphs, the authors showed that although the eigenvalue-based approximation is reasonable for both the large and the small N limits, it fails to accurately predict the learning curve in the important transition region between these two extremes, drastically so for low noise variances σ^2. We will show that this shortcoming can be overcome by taking advantage of the sparse structure of the underlying graph using the cavity method.

3 Accurate predictions with the cavity method

The cavity method was developed in statistical physics [13] but is closely related to belief propagation. We begin with equation (3). Because we only need the posterior variance in the matched case considered here, we can shift $\boldsymbol{f}$ so that $\bar{\boldsymbol{f}} = \mathbf{0}$; f_i is then the deviation of the function value at node i from the posterior mean. In this notation, the Bayes error is

$$\epsilon = \langle \frac{1}{V} \sum_i \int d\boldsymbol{f} f_i^2 P(f|D) \rangle_{D,\text{graphs}} \tag{5}$$

where $P(f|D)$ now contains in the exponent only the terms from (1) that are quadratic in $\boldsymbol{f}$.

To set up the cavity method, we begin by defining a *generating* or *partition function* Z, for a fixed graph, as

$$Z = \int d\boldsymbol{f} \exp(-\frac{1}{2}\boldsymbol{f}^T \boldsymbol{C}^{-1} \boldsymbol{f} - \frac{1}{2\sigma^2} \sum_\mu f_{i_\mu}^2 - \frac{\lambda}{2} \sum_i f_i^2) \tag{6}$$

An auxiliary parameter λ has been added here to allow us to represent the Bayes error as $\epsilon = -\lim_{\lambda \to 0} (2/V) \frac{\partial}{\partial \lambda} \langle \log Z \rangle_{D,\text{graphs}}$. The dependence on the dataset D appears in Z only through the sum over μ. It will be more useful to write this as a sum over all nodes: if n_i counts the number of examples seen at node i, then $\sum_\mu f_{i_\mu}^2 = \sum_i n_i f_i^2$. Even with this replacement, the partition function in equation (6) is not yet suitable for an application of the cavity method since the inverse covariance function cannot be written explicitly and generates interaction terms $f_i f_j$ between nodes that can be far away from each other along the graph. To eliminate the inverse of the covariance function we therefore perform a Fourier transform on the first term in the exponent, $\exp(-\frac{1}{2}\boldsymbol{f}^T \boldsymbol{C}^{-1} \boldsymbol{f}) \propto \int d\boldsymbol{h} \exp(-\frac{1}{2}\boldsymbol{h}^T \boldsymbol{C} \boldsymbol{h} + i \sum_i h_i f_i)$. The integral over $\boldsymbol{f}$ then factorizes over the f_i, and one finds

$$Z \propto \int d\boldsymbol{h} \exp(-\frac{1}{2}\boldsymbol{h}^T \boldsymbol{C} \boldsymbol{h} - \frac{1}{2}\boldsymbol{h}^T \text{diag}\{(\frac{n_i}{\sigma^2} + \lambda)^{-1}\} \boldsymbol{h}) \tag{7}$$

Substituting the explicit form of the covariance function (2) into equation (7) we have

$$Z \propto \int d\boldsymbol{h} \exp(-\frac{1}{2}\boldsymbol{h}^T \sum_{q=0}^{p} c_q (\boldsymbol{D}^{-1/2} \boldsymbol{A} \boldsymbol{D}^{-1/2})^q \boldsymbol{h} - \frac{1}{2}\boldsymbol{h}^T \text{diag}\{(\frac{n_i}{\sigma^2} + \lambda)^{-1}\} \boldsymbol{h}) \tag{8}$$

where we have written the power in equation (2) as a binomial sum and defined $c_q = p!/[q!(p-q)!] a^{-q} (1 - a^{-1})^{p-q} / \kappa$.

For $p > 1$, equation (8) still has interactions with more than the immediate neighbours. To solve this we introduce additional variables $\boldsymbol{h}^q$, defined recursively via $\boldsymbol{h}^q = (\boldsymbol{D}^{-1/2} \boldsymbol{A} \boldsymbol{D}^{-1/2}) \boldsymbol{h}^{q-1}$ for $q \geq 1$ and $\boldsymbol{h}^0 = \boldsymbol{h}$. These definitions are enforced via Dirac delta-functions, each i and $q \geq 1$ giving a factor $\delta(h_i^q - d_i^{-1/2} \sum_j A_{ij} d_j^{-1/2} h_j^{q-1}) \propto \int d\hat{h}_i^q \exp[i \hat{h}_i^q (h_i^q - d_i^{-1/2} \sum_j A_{ij} d_j^{-1/2} h_j^{q-1})]$. Substituting this into equation (8) gives the key advantage that now the adjacency matrix appears only linearly in the exponent, so that we have interactions only across edges of the graph. Rescaling the h_i^q to $d_i^{1/2} h_i^q$ and similarly for the $\hat{h}_i^q$, and explicitly separating off the local terms from the interactions finally yields

$$\begin{aligned} Z \propto \int \prod_{q=0}^{p} d\boldsymbol{h}^q \prod_{q=1}^{p} d\hat{\boldsymbol{h}}^q \prod_i \exp(-\frac{1}{2} \sum_{q=0}^{p} c_q d_i h_i^0 h_i^q - \frac{1}{2} \frac{d_i (h_i^0)^2}{n_i/\sigma^2 + \lambda} + i \sum_{q=1} d_i \hat{h}_i^q h_i^q) \\ \times \prod_{(ij)} \exp(-i \sum_{q=1} (\hat{h}_i^q h_j^{q-1} + \hat{h}_j^q h_i^{q-1})) \end{aligned} \tag{9}$$

We now have the partition function of a (complex-valued) Gaussian graphical model. By differentiating $\log Z$ with respect to λ, keeping track of λ-dependent prefactors not written above, one finds that the Bayes error is,

$$\epsilon = \lim_{\lambda \to 0} \frac{1}{V} \sum_i \frac{1}{n_i/\sigma^2 + \lambda} \left(1 - \frac{d_i \langle (h_i^0)^2 \rangle}{n_i/\sigma^2 + \lambda} \right) \tag{10}$$

and so we need the marginal distributions of the h_i^0. This is where the cavity method enters: for a large random graph the structure is locally treelike, so that if node i were eliminated the corresponding subgraphs (locally trees) rooted at the neighbours $j \in \mathcal{N}(i)$ of i would become independent [12]. The resulting cavity marginals $P_j^{(i)}(\boldsymbol{h}_j, \hat{\boldsymbol{h}}_j | D)$ can then be calculated iteratively within these subgraphs, giving the cavity update equations

$$\begin{aligned} P_j^{(i)}(\boldsymbol{h}_j, \hat{\boldsymbol{h}}_j | D) \propto \exp(-\frac{1}{2} \sum_{q=0}^{p} c_q d_j h_j^0 h_j^q - \frac{1}{2} \frac{d_j (h_j^0)^2}{n_j/\sigma^2 + \lambda} + i \sum_{q=1}^{p} d_j \hat{h}_j^q h_j^q) \\ \int \prod_{k \in \mathcal{N}(j) \setminus i} d\boldsymbol{h}_k d\hat{\boldsymbol{h}}_k \, \exp(-i \sum_{q=1}^{p} (\hat{h}_j^q h_k^{q-1} + \hat{h}_k^q h_j^{q-1})) P_k^{(j)}(\boldsymbol{h}_k, \hat{\boldsymbol{h}}_k | D) \end{aligned} \tag{11}$$

One sees that these equations are solved self-consistently by complex-valued Gaussian distributions with mean zero and covariance matrices $\boldsymbol{V}_j^{(i)}$. By performing the Gaussian integrals in the cavity update equations (11) explicitly, these equations then take the rather simple form

$$\boldsymbol{V}_j^{(i)} = (\boldsymbol{O}_j - \sum_{k \in \mathcal{N}(j) \setminus i} \boldsymbol{X} \boldsymbol{V}_k^{(j)} \boldsymbol{X})^{-1} \tag{12}$$

where we have defined the $(2p+1) \times (2p+1)$ matrices

$$\boldsymbol{O}_i = d_i \left(\begin{array}{cccc|ccc} c_0 + \frac{1}{n_i/\sigma^2 + \lambda} & \frac{1}{2}c_1 & \dots & \frac{1}{2}c_p & 0 & \dots & 0 \\ \frac{1}{2}c_1 & & & & -i & & \\ \vdots & & & & & \ddots & \\ \frac{1}{2}c_p & & & & & & -i \\ \hline 0 & -i & & & & & \\ \vdots & & \ddots & & & \boldsymbol{0}_{p,p} & \\ 0 & & & -i & & & \end{array} \right), \quad \boldsymbol{X} = \left(\begin{array}{ccc|ccc} & & & i & & \\ & \boldsymbol{0}_{p+1,p+1} & & & \ddots & \\ & & & & & i \\ & & & 0 & \dots & 0 \\ \hline i & & 0 & & & \\ & \ddots & \vdots & & \boldsymbol{0}_{p,p} & \\ & i & 0 & & & \end{array} \right)$$

Finally we need to translate these equations to an ensemble of large sparse graphs. Each ensemble is characterised by the distribution $p(d)$ of the degrees d_i, with every graph having the desired degree distribution being assigned the same probability. Instead of individual cavity covariance matrices $\boldsymbol{V}_j^{(i)}$, we need to consider their probability distribution $W(\boldsymbol{V})$ across all edges of the graph. Picking at random an edge (i, j) of a graph, the probability that node j will have degree d_j is then $p(d_j) d_j / \bar{d}$, because such a node has d_j "chances" of being picked. (The normalisation factor is the average degree $\bar{d}$.) Using again the locally treelike structure, the incoming (to node j) cavity covariances $\boldsymbol{V}_k^{(j)}$ will be i.i.d. samples from $W(\boldsymbol{V})$. Thus a fixed point of the cavity update equations corresponds to a fixed point of an update equation for $W(\boldsymbol{V})$:

$$W(\boldsymbol{V}) = \left\langle \sum_d \frac{p(d) d}{\bar{d}} \int \prod_{k=1}^{d-1} d\boldsymbol{V}_k \, W(\boldsymbol{V}_k) \, \delta(\boldsymbol{V} - (\boldsymbol{O} - \sum_{k=1}^{d-1} \boldsymbol{X} \boldsymbol{V}_k \boldsymbol{X})^{-1}) \right\rangle_n \tag{13}$$

Because the node label is now arbitrary, we have abbreviated $\boldsymbol{V}_j^{(i)}$ to $\boldsymbol{V}$, d_j to d, $\boldsymbol{O}_j$ to $\boldsymbol{O}$ and $\boldsymbol{V}_k^{(j)}$ to $\boldsymbol{V}_k$. The average is over the distribution over the number of examples $n \equiv n_j$ at node j in the dataset D. Assuming for simplicity that examples are drawn with uniform input probability across all nodes, this distribution is simply $n \sim \text{Poisson}(\nu)$ in the limit of large N and V at fixed $\nu = N/V$.

In general equation (13) – which can also be formally derived using the replica approach [17] – cannot be solved analytically, but we can solve it numerically using a standard population dynamics

method [18]. Once we have $W(\boldsymbol{V})$, the Bayes error can be found from the graph ensemble version of equation (10), which is obtained by inserting the explicit expression for $\langle (h_i^0)^2 \rangle$ in terms of the cavity marginals of the neighbouring nodes, and replacing the average over nodes with an average over $p(d)$:

$$\epsilon = \lim_{\lambda \to 0} \left\langle \sum_d \frac{p(d)}{n/\sigma^2 + \lambda} \left(1 - \frac{d}{n/\sigma^2 + \lambda} \int \prod_{k=1}^{d} d\boldsymbol{V}_k \, W(\boldsymbol{V}_k) \, (\boldsymbol{O} - \sum_{k=1}^{d} \boldsymbol{X}\boldsymbol{V}_k\boldsymbol{X})^{-1}_{00} \right) \right\rangle_n \tag{14}$$

The number of examples at the node is again to be averaged over $n \sim \text{Poisson}(\nu)$. The subscript "00" indicates the top left element of the matrix, which determines the variance of h^0.

To be able to use equation (14), it needs to be rewritten in a form that remains explicitly non-singular for $n = 0$. We split off the n-dependence of the matrix inverse by writing $\boldsymbol{O} - \sum_{k=1}^{d} \boldsymbol{X}\boldsymbol{V}_k\boldsymbol{X} = \boldsymbol{M} + [d/(n/\sigma^2 + \lambda)]\boldsymbol{e}_0\boldsymbol{e}_0^T$, where $\boldsymbol{e}_0^T = (1, 0, \ldots, 0)$. The matrix inverse appearing above can then be expressed using the Woodbury formula as

$$\boldsymbol{M}^{-1} - \frac{\boldsymbol{M}^{-1}\boldsymbol{e}_0\boldsymbol{e}_0^T\boldsymbol{M}^{-1}}{(n/\sigma^2 + \lambda)/d + \boldsymbol{e}_0^T\boldsymbol{M}^{-1}\boldsymbol{e}_0} \tag{15}$$

To extract the (0,0)-element (top left) as required we multiply by $\boldsymbol{e}_0^T \cdots \boldsymbol{e}_0$. After some simplification the $\lambda \to 0$ limit can then be taken, with the result

$$\epsilon = \left\langle \sum_d p(d) \int \prod_{k=1}^{d} d\boldsymbol{V}_k \, W(\boldsymbol{V}_k) \, \frac{1}{n/\sigma^2 + d(\boldsymbol{M}^{-1})_{00}} \right\rangle_n \tag{16}$$

This has a simple interpretation: the cavity marginals of the neighbours provide an effective Gaussian prior for each node, whose inverse variance is $d(\boldsymbol{M}^{-1})_{00}$.

The self-consistency equation (13) for $W(\boldsymbol{V})$ and the expression (16) for the resulting Bayes error are our main results so far. They allow us to predict learning curves as a function of the number of examples per node, ν, for *arbitrary degree distributions* $p(d)$ of our random graph ensemble providing the graphs are sparse and for arbitrary noise level σ^2 and covariance function hyperparameters p and a.

4 Generalising to community structures

The graphs considered in section 3 do not lend themselves to representing community structure. Fixing the degree distribution alone means that all nodes have interchangeable roles, and cannot account for the large number of intra-community connections compared to the smaller number of inter-community connections. More importantly, communities tend to contain many short loops, and these are neglected in the calculation of the single node cavity marginals considered in the previous section.

To resolve this we consider graphs generated from ensembles similar to those studied in [14, 15]. We assume graphs are generated by a sparse superstructure a_{super} with a fixed arbitrary degree distribution. This superstructure then governs which communities are connected to each other. Which nodes from the two communities are involved in such interconnections will be encoded by a distribution $\mu(\boldsymbol{A}_{\text{inter}})$ and local connections within communities by a distribution $\rho(\boldsymbol{A}_{\text{intra}})$. We consider communities of fixed size M so that the A-matrices here are $M \times M$. Thus given a superstructure $\{a_{ij}\}$ randomly generated from a degree distribution $p(d)$ the graph is sampled from the distribution,

$$P(\boldsymbol{A}|\{a_{ij}\}) = \prod_{i<j} \left(\frac{\bar{d}}{V} \delta_{a_{ij},1} \mu(\boldsymbol{A}^{ij}_{\text{inter}}) + (1 - \frac{\bar{d}}{V}) \delta_{a_{ij},0} \delta(\boldsymbol{A}_{\text{inter}}) \right) \prod_i \rho(\boldsymbol{A}_{\text{intra}}) \tag{17}$$

In order to calculate predictions of the generalisation error ϵ_g using the method in section 3 we must treat h_i in equation (9) as a vector $\boldsymbol{h}_i$ of the Fourier transformed function values of all the nodes contained in a community i. We can then use the sparsity of the superstructure and get accurate marginals for a *whole community*. Proceeding as before but with $(2p+1)M \times (2p+1)M$ variance

matrices for each community we obtain the update equations,

$$W(\boldsymbol{V}) = \left\langle \sum_d \frac{p(d)d}{\bar{d}} \int \prod_{i=1}^{d-1} d\boldsymbol{V}^i \, W(\boldsymbol{V}^i) \sum_{\boldsymbol{A}^1_{\text{inter}} \dots \boldsymbol{A}^d_{\text{inter}}} \mu(\boldsymbol{A}^1_{\text{inter}}) \dots \mu(\boldsymbol{A}^d_{\text{inter}}) \sum_{\boldsymbol{A}_{\text{intra}}} \rho(\boldsymbol{A}_{\text{intra}}) \right. \\ \left. \delta(\boldsymbol{V} - (\boldsymbol{O} - \sum_{k=1}^{d-1} \boldsymbol{X}^k \boldsymbol{V}^k \boldsymbol{X}^k)^{-1}) \right\rangle_{n_1,\dots,n_M} \tag{18}$$

where we have defined $M \times M$ matrices $\boldsymbol{D} = \text{diag}_{k=1\dots M}\{\sum_l((\boldsymbol{A}_{\text{intra}})_{kl} + \sum_{i=1}^d((\boldsymbol{A}_{\text{inter}})^i)_{kl})\}$ and $\boldsymbol{S} = \text{diag}_{i=1\dots M}\{\frac{1}{n_i/\sigma^2+\lambda}\}$, and $(2p+1)M \times (2p+1)M$ matrices

$$\boldsymbol{X}^i = \left(\begin{array}{ccc|ccc} & & & i\boldsymbol{A}^i_{\text{inter}} & & \\ & \mathbf{0}_{(p+1)M,(p+1)M} & & & \ddots & \\ & & & & & i\boldsymbol{A}^i_{\text{inter}} \\ & & & 0 & \dots & 0 \\ \hline i(\boldsymbol{A}^i_{\text{inter}})^T & & 0 & & & \\ & \ddots & \vdots & & \mathbf{0}_{pM,pM} & \\ & i(\boldsymbol{A}^i_{\text{inter}})^T & 0 & & & \end{array}\right)$$

$$\boldsymbol{O} = \left(\begin{array}{ccccc|ccc} c_0\boldsymbol{D}+\boldsymbol{D}\boldsymbol{S} & \frac{1}{2}c_1\boldsymbol{D} & \dots & \frac{1}{2}c_p\boldsymbol{D} & & i\boldsymbol{A}_{\text{intra}} & 0 & \dots & 0 \\ \frac{1}{2}c_1\boldsymbol{D} & & & & & -i\boldsymbol{D} & \ddots & & \\ \vdots & & & & & & \ddots & & i\boldsymbol{A}_{\text{intra}} \\ \frac{1}{2}c_p\boldsymbol{D} & & & & & & & & -i\boldsymbol{D} \\ \hline i\boldsymbol{A}_{\text{intra}} & -i\boldsymbol{D} & & & & & & & \\ 0 & \ddots & \ddots & & & & \mathbf{0}_{pM,pM} & & \\ \vdots & & & & & & & & \\ 0 & & i\boldsymbol{A}_{\text{intra}} & -i\boldsymbol{D} & & & & & \end{array}\right)$$

Solving for $W(\boldsymbol{V})$ as before with population dynamics we can then use the cavity marginals to produce the required full marginals for the generalisation error,

$$\epsilon = \lim_{\lambda\to 0} \frac{1}{M} \sum_{m=1}^{M} \left[\left\langle \sum_d p(d) \boldsymbol{D}\boldsymbol{S} \left(\boldsymbol{I} - \boldsymbol{D}\boldsymbol{S} \int \prod_{k=1}^{d} d\boldsymbol{V}^k \, W(\boldsymbol{V}^k) \sum_{\boldsymbol{A}^1_{\text{inter}} \dots \boldsymbol{A}^d_{\text{inter}}} \right. \right. \right. \\ \left. \left. \left. \mu(\boldsymbol{A}^1_{\text{inter}}) \dots \mu(\boldsymbol{A}^d_{\text{inter}}) \sum_{\boldsymbol{A}_{\text{intra}}} \rho(\boldsymbol{A}_{\text{intra}}) (\boldsymbol{O} - \sum_{k=1}^{d} \boldsymbol{X}^k \boldsymbol{V}^k \boldsymbol{X}^k)^{-1}_{00} \right) \right\rangle_{n_1 \dots n_M} \right]_{mm} \tag{19}$$

Again this has an apparent singularity when any of the $n_m = 0$ in a community, which can be removed by using the Woodbury formula as in section 3. We emphasise once more that our cavity marginals pass variances for a whole community. This results in the additional average over nodes inside the community seen in equation (19).

5 Results

We will begin by comparing the performance of our new cavity prediction (equation (16)) against the eigenvalue approximation (equation (4)) from [4, 7] for a Poisson random graph, $p(d) = c^d e^{-c}/d!$, for $c = 3$ with 500 nodes.

As can be seen in figure 1 (left) the cavity approach is accurate along the entire learning curve, to the point where the prediction is visually almost indistinguishable from the numerical simulation results. Importantly, the cavity approach predicts even the midsection of the learning curve for intermediate values of ν, where the eigenvalue prediction clearly fails. The deviations between cavity theory

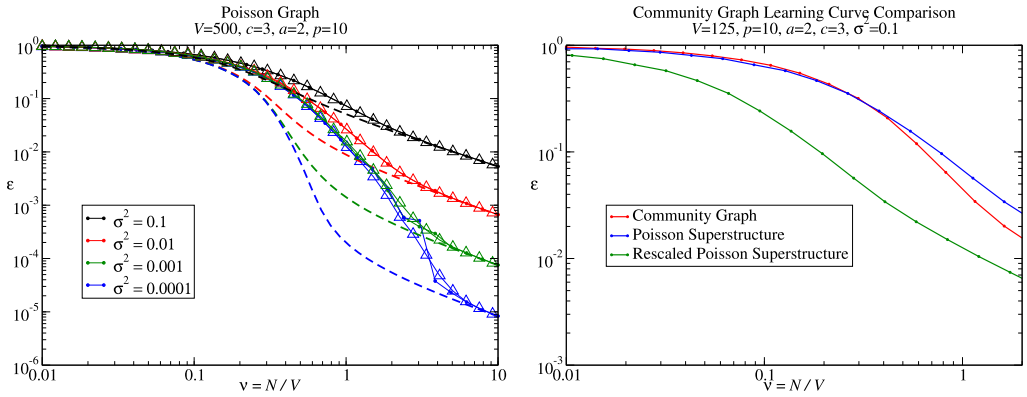

Figure 1: (left) A comparison of the cavity prediction (solid line with triangles) against the eigenvalue approximation (dashed line) for the learning curves for Poisson graphs of average degree $c = 3$, and against simulation results for graphs with $V = 500$ nodes (solid line with circles). Random walk kernel with $p = 10$, $a = 2$; noise level as shown. (right) Numerical learning curves of a community graph (red) with Poisson superstructure (with $c = 3$), all connections present within communities of size $M = 4$ and a single randomly chosen edge making the connection between any two communities linked in the superstructure ($V = 125$). These are compared to results for sparse graphs with the same connectivity as the community superstructure and $V = 125$ nodes (blue), and Poisson connectivity $c = 3$ for $V = 125$ rescaled so absolute number of examples is compared (green).

and the eigenvalue predictions are largest in this central part because at this point fluctuations in the number examples seen at each node have the greatest effect. Indeed, for much smaller ν, the dataset does not contain any examples from many of the nodes, i.e. $n = 0$ is dominant and fluctuations towards larger n have low probability. For large ν, the dataset typically contains many examples for each node and Poisson fluctuations around the average value $n = \nu$ are small. The fluctuation effects for intermediate ν are suppressed when the noise level σ^2 is large, because then the generalisation error in the range of intermediate ν is still fairly close to its initial value ($\nu = 0$). But for the smaller noise levels fluctuations in the number of examples for each node can have a large effect, and correspondingly the eigenvalue prediction becomes very poor for intermediate ν.

Finally figure 1 (right) shows a numerical learning curve for a community graph ensemble with a Poisson random graph superstructure with $c = 3$, linking communities of size $M = 4$. Communities are taken as fully connected internally, with inter-community connections in the superstructure taken as single edges connecting two randomly chosen nodes from each of the communities. Figure 1 (right) compares the community learning curve against two baselines without community structure, i.e. sparse graphs. We compare with a Poisson random graph of just the superstructure, where effectively each community is replaced by a single node and a rescaled version of the superstructure random graph comparing absolute number of examples. As can be seen learning on community ensembles is drastically different from models enforcing only degree distributions. Community learning curves begin by approximately following the superstructure learning curve then tend towards the rescaled superstructure curves as examples increase. We believe this is because for smaller numbers of examples the community learning curve is dominated by examples on nodes that are uncorrelated with each other, community structure therefore helps very little. For higher numbers of examples the GP is increasingly able to infer from one nodes example the value of other nodes within the highly connected community. In essence an example seen in the community allows the GP to infer the entire community, this is equivalent to the information gained about a single node by seeing an example of that node in the superstructure graph and thus the community learning curve tends to the rescaled version as examples increase. It is worth noting that covariance parameters have been kept constant in this comparison, this will result in non-trivial changes in the range and locality of covariance matrix superstructure. Learning curves will have less local, further reaching kernels.

One would expect predictions using the learning curves in section 4 to generate far more accurate results than predictions for the other two graph ensembles, and we will be able to confirm this by the time of the workshop.

6 Conclusions and further work

In this paper we have studied the learning curves of GP regression on large random graphs. In a significant advance on the work of [10], we showed that the approximations for learning curves proposed by Sollich [4] and Opper [7] for continuous input spaces can be greatly improved upon in the graph case, by using the cavity method. We argued that the resulting predictions should in fact become exact in the limit of large random graphs.

Section 3 derived the learning curve approximation using the cavity method for *arbitrary degree distributions*. In section 4 we generalised our new prediction to community graph ensembles where we assume a sparse superstructure connecting communities consisting of a fixed number of nodes each. By grouping nodes into their communities and applying the cavity method in section 3 to these grouped nodes we were able to derive analogous update equations (equation (18)) and generalisation error predictions (equation (19)).

Finally in section 5 we compared our generalised error predictions for an arbitrary fixed degree distribution to an older eigenvalue prediction seen in Sollich et. al. [10] and showed a vast improvement in predicting these learning curves in particular in the challenging midsection of these curves. Further we showed in figure 1 (right) that community graph ensembles cannot be modelled by the degree distribution ensembles in section 3 and require the more general generalisation error predictions given in section 4.

We would like to extend the community derived learning curves in section 4 to more general community sizes. Introducing a distribution of community sizes and correspondingly a conditional distribution of intra and inter connections one would be able to derive learning curve predictions for a more general class of community graph ensembles. It also would be interesting to introduce degree-degree correlations as seen in [15] this would enable more control over the structure of the sparse graphs considered in this paper.

We think it would be interesting to expand our approach to model mismatch, where we assume the data-generating process is a GP with hyperparameters that differ from those of the GP being used for inference. It should further be useful to study the case of mismatched *graphs*, rather than hyperparameters. Finally, it would be worth extending the study of graph mismatch to the case of evolving graphs and functions.

References

[1] Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, December 2005.

[2] Shun-ichi Amari, Naotake Fujita, and Shigeru Shinomoto. Four types of learning curves. *Neural Computation*, 4(4):605–618, 1992.

[3] M. Opper. Regression with Gaussian processes: Average case performance. *Theoretical Aspects of Neural Computation: A Multidisciplinary Perspective. Springer-Verlag*, pages 17–23, 1997.

[4] P. Sollich. Learning curves for Gaussian processes. In S A Solla M S Kearns and D A Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 344–350. MIT Press, 1999.

[5] F. Vivarelli and M. Opper. General bounds on Bayes errors for regression with Gaussian processes. In S A Solla M S Kearns and D A Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 302–308. MIT Press, 1999.

[6] C. K. I. Williams and F. Vivarelli. Upper and lower bounds on the learning curve for Gaussian processes. *Machine Learning*, 40(1):77–102, 2000.

[7] M. Opper and D. Malzahn. Learning curves for gaussian processes regression: A framework for good approximations. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 273–279. MIT Press, 2001.

[8] P. Sollich and A. Halees. Learning curves for Gaussian process regression: Approximations and bounds. *Neural Computation*, 14(6):1393–1428, 2002.

[9] P. Sollich. Gaussian process regression with mismatched models. In S Becker T G Dietterich and Z Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 519–526. MIT Press, 2002.

[10] P. Sollich, M. J. Urry, and C. Coti. Kernels and learning curves for Gaussian process regression on random graphs. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1723–1731. Curran Associates, Inc., 2009.

[11] M. Opper and D. Malzahn. A variational approach to learning curves. In *Advances in Neural Information Processing Systems 14*, pages 463–469. MIT Press, 2002.

[12] Tim Rogers, Koujin Takeda, Issac Pérez Castillo, and Reimer Kühn. Cavity approach to the spectral density of sparse symmetric random matricies. *Physical Review E*, 78(3):31116–31121, 2008.

[13] M. Mezard, G. Parisi, and M. A. Virasoro. Random free energies in spin glasses. *Le journal de physique - lettres*, 46(6):217–222, 1985.

[14] Ginestra Bianconi, Anthony C. C. Coolen, and Conrad J. Perez Vicente. Entropies of complex networks with hierarchically constrained topologies. *Phys. Rev. E*, 78(1):016114, Jul 2008.

[15] Tim Rogers, Conrad Pérez Vicente, Koujin Takeda, and Isaac Pérez Castillo. Spectral density of random graphs with topological constraints. *Journal of Physics A*, 43(19):195002, 2010.

[16] A. J. Smola and R. Kondor. Kernels and regularization on graphs. In M. Warmuth and B. Scholkopf, editors, *Learning theory and Kernel machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop (COLT)*, pages 144–158, Heidelberg, 2003. Springer.

[17] Reimer Kühn. Finitely coordinated models for low-temperature phases of amorphous systems. *Journal of Physics A*, 40(31):9227, 2007.

[18] M. Mézard and G. Parisi. The Bethe lattice spin glass revisited. *The European Physical Journal B*, 20(2):217–233, 2001.