Random Walk Kernels and Learning Curves for Gaussian Process Regression on Random Graphs

Matthew J. Urry   MATTHEW.URRY@KCL.AC.UK
Peter Sollich   PETER.SOLLICH@KCL.AC.UK
Department of Mathematics
King's College London
London, WC2R 2LS, U.K.

Editor: Manfred Opper

Abstract

We consider learning on graphs, guided by kernels that encode similarity between vertices. Our focus is on random walk kernels, the analogues of squared exponential kernels in Euclidean spaces. We show that on large, locally treelike graphs these have some counter-intuitive properties, specifically in the limit of large kernel lengthscales. We consider using these kernels as covariance functions of Gaussian processes. In this situation one typically scales the prior globally to normalise the average of the prior variance across vertices. We demonstrate that, in contrast to the Euclidean case, this generically leads to significant variation in the prior variance across vertices, which is undesirable from a probabilistic modelling point of view. We suggest the random walk kernel should be normalised locally, so that each vertex has the same prior variance, and analyse the consequences of this by studying learning curves for Gaussian process regression. Numerical calculations as well as novel theoretical predictions for the learning curves using belief propagation show that one obtains distinctly different probabilistic models depending on the choice of normalisation. Our method for predicting the learning curves using belief propagation is significantly more accurate than previous approximations and should become exact in the limit of large random graphs.

Keywords: Gaussian process, generalisation error, learning curve, cavity method, belief propagation, graph, random walk kernel

1. Introduction

Gaussian processes (GPs) have become a workhorse for probabilistic inference that has been developed in a wide range of research fields under various guises (see for example Kleijnen, 2009; Handcock and Stein, 1993; Neal, 1996; Meinhold and Singpurwalla, 1983). Their success and wide adoption can be attributed mainly to their intuitive nature and ease of use. They owe their intuitiveness to being one of a large family of kernel methods that implicitly map lower dimensional spaces with non-linear relationships to higher dimensional spaces where (hopefully) relationships are linear. This feat is achieved by using a kernel, which also encodes the types of functions that the GP prefers a priori. The ease of use of GPs is due to the simplicity of implementation, at least in the basic setting, where prior and posterior distributions are both Gaussian and can be written explicitly.

An important question for any machine learning method is how 'quickly' the method can generalise its prediction of a rule to the entire domain of the rule (i.e., how many examples are required to achieve a particular generalisation error). This is encapsulated in the *learning curve*, which traces average error versus number of examples. Learning curves have been studied for a variety of infer-

ence methods and are well understood for parametric models (Seung et al., 1992; Amari et al., 1992; Watkin et al., 1993; Opper and Haussler, 1995; Haussler et al., 1996; Freeman and Saad, 1997) but rather less is known for non-parametric models such as GPs. In the case of GP regression, research has predominantly focused on leaning curves for input data from Euclidean spaces (Sollich, 1999a,b; Opper and Vivarelli, 1999; Williams and Vivarelli, 2000; Malzahn and Opper, 2003; Sollich, 2002; Sollich and Halees, 2002; Sollich and Williams, 2005), but there are many domains for which the input data has a discrete structure. One of the simplest cases is the one where inputs are vertices on a graph, with connections on the graph encoding similarity relations between different inputs. Examples could include the internet, social networks, protein networks and financial markets. Such discrete input spaces with graph structure are becoming more important, and therefore so is an understanding of GPs, and machine learning techniques in general, on these spaces.

In this paper we expand on earlier work in Sollich et al. (2009) and Urry and Sollich (2010) and focus on predicting the learning curves of GPs used for regression (where outputs are from the whole real line) on large sparse graphs, using the *random walk kernel* (Kondor and Lafferty, 2002; Smola and Kondor, 2003).

The rest of this paper will be structured as follows. In Section 2 we begin by analysing the random walk kernel, in particular with regard to the dependence on its lengthscale parameter, and study the approach to the fully correlated limit. With a better understanding of the random walk kernel in hand, we proceed in Section 3 to an analysis of the use of the random walk kernel for GP regression on graphs. We begin in Section 3.2 by looking at how kernel normalisation affects the prior probability over functions. We show that the more frequently used global normalisation of a kernel by its average prior variance is inappropriate for the highly location dependent random walk kernel, and suggest normalisation to uniform local prior variance as a remedy. To understand how this affects GP regression using random walk kernels quantitatively, we extend first in Section 3.4 an existing approximation to the learning curve in terms of kernel eigenvalues (Sollich, 1999a; Opper and Malzahn, 2002) to the discrete input case, allowing for arbitrary normalisation. This approximation turns out to be accurate only in the initial and asymptotic regimes of the learning curve.

The core of our analysis begins in Section 4 with the development of an improved learning curve approximation based on belief propagation. We first apply this, in Section 4.1, to the case of globally normalised kernels as originally proposed. The belief propagation analysis for global normalisation also acts as a useful warm-up for the extension to the prediction of learning curves for the locally normalised kernel setting, which we present in Section 4.3. In both sections we compare our predictions to numerical simulations, finding good agreement that improves significantly on the eigenvalue approximation. Finally, to emphasise the distinction between the use of globally and locally normalised kernels in GP regression, we study qualitatively the case of model mismatch, with a GP with a globally normalised kernel as the teacher and a GP with a locally normalised kernel as the student, or visa versa. The resulting learning curves show that the priors arising from the two different normalisations are fundamentally different; the learning curve can become non-monotonic and develop a maximum as a result of the mismatch. We conclude in Section 6 by summarising our results and discussing further potentially fruitful avenues of research.

1.1 Main Results

In this paper we will derive three key results; that normalisation of a kernel by its average prior variance leads to a complicated relationship between the prior variances and the local graph structure; that by fixing the scale to be equal everywhere using a local prescription $C_{ij} = \hat{C}_{ij}/\sqrt{\hat{C}_{ii}\hat{C}_{jj}}$ results in a fundamentally different probabilistic model; and that we can derive accurate predictions of the learning curves of Gaussian processes on graphs with a random walk kernel for both normalisations over a broad range of graphs and parameters. The last result is surprising since in continuous spaces this is only possible for a few very restrictive cases.

2. The Random Walk Kernel

A wide range of machine learning techniques like Gaussian processes capture prior correlations between points in an input space by mapping to a higher dimensional space, where correlations can be represented by a linear combination of 'features' (see, e.g., Rasmussen and Williams, 2005; Müller et al., 2001; Cristianini and Shawe-Taylor, 2000). Direct calculation of correlations in this high dimensional space is avoided using the 'kernel trick', where the kernel function implicitly calculates inner products in feature space. The widespread use of, and therefore extensive research in, kernel based machine learning has resulted in kernels being developed for a wide range of input spaces (see Genton, 2002, and references therein). We focus in this paper on the class of kernels introduced in Kondor and Lafferty (2002). These make use of the normalised graph Laplacian to define correlations between vertices of a graph.

We denote a generic graph by $G(\mathcal{V}, \mathcal{E})$ with a vertex set $\mathcal{V} = \{1, \dots, V\}$ and edge set $\mathcal{E}$. We encode the connection structure of G using an adjacency matrix $\boldsymbol{A}$, where $A_{ij} = 1$ if vertex i is connected to j, and 0 otherwise; we exclude self-loops so that $A_{ii} = 0$. We denote the number of edges connected to vertex i, known as the degree, by $d_i = \sum_j A_{ij}$ and define the degree matrix $\boldsymbol{D}$ as a diagonal matrix of the vertex degrees, that is, $D_{ij} = d_i \delta_{ij}$. The class of kernels created in Kondor and Lafferty (2002) is constructed using the *normalised Laplacian*, $\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$ (see Chung, 1996) as a replacement for the Laplacian in continuous spaces. Of particular interest is the diffusion kernel and its easier to calculate approximation, the random walk kernel. Both of these kernels can be viewed as an approximation to the ubiquitous squared exponential kernel that is used in continuous spaces. The direct graph equivalent of the squared exponential kernel is given by the *diffusion kernel* (Kondor and Lafferty, 2002). It is defined as

$$C = \exp\left(-\frac{1}{2}\sigma^2 \boldsymbol{L}\right), \quad \sigma > 0, \tag{1}$$

where σ sets the length-scale of the kernel. Unlike in continuous spaces, the exponential in the diffusion kernel is costly to calculate. To avoid this, Smola and Kondor (2003) proposed as a cheaper approximation the *random walk kernel*

$$C = \left(\boldsymbol{I} - a^{-1}\boldsymbol{L}\right)^p = \left((1 - a^{-1})\boldsymbol{I} + a^{-1}\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}\right)^p, \quad a > 2, \quad p \in \mathbb{N}. \tag{2}$$

This gives back the diffusion kernel in the limit $a, p \to \infty$ whilst keeping $p/a = \sigma^2/2$ fixed. The random walk kernel derives its name from its use of random walks to express correlations between

vertices. Explicitly, a binomial expansion of Equation (2) gives

$$
\begin{aligned}
C &= \sum_{q=0}^{p} \binom{p}{q} (1 - a^{-1})^{p-q} (a^{-1})^q (D^{-1/2} A D^{-1/2})^q \\
&= D^{-1/2} \sum_{q=0}^{p} \binom{p}{q} (1 - a^{-1})^{p-q} (a^{-1})^q (A D^{-1})^q D^{1/2}.
\end{aligned}
\tag{3}
$$

The matrix AD^{-1} is a random walk transition matrix: $(AD^{-1})_{ij}$ is the probability of being at vertex i after one random walk step starting from vertex j. Apart from the pre- and post-multiplication by $D^{-1/2}$ and $D^{1/2}$, the kernel C is therefore a q-step random walk transition matrix, averaged over the number of steps q distributed as $q \sim \text{Binomial}(p, a^{-1})$. Equivalently one can interpret the random walk kernel as a p-step lazy random walk, where at each step the walker stays at the current vertex with probability $(1 - a^{-1})$ and moves to a neighbouring vertex with probability a^{-1}.

Using either interpretation, one sees that p/a is the lengthscale over which the random walk can diffuse along the graph, and hence the lengthscale describing the typical maximum range of the random walk kernel. In the limit of large p, where this lengthscale diverges, the kernel should represent full correlation across all vertices. One can see that this is the case by observing that for large p, a random walk on a graph will approach its stationary distribution, $p_\infty \propto De$, $e = (1, \dots, 1)^T$. The q-step transition matrix for large q is therefore $(AD^{-1})^q \approx p_\infty e^T = Dee^T$, representing the fact that the random walk becomes stationary independently of the starting vertex. This gives, for $p \to \infty$, the kernel $C \propto D^{1/2} ee^T D^{1/2}$, that is, $C_{ij} \propto d_i^{1/2} d_j^{1/2}$. This corresponds to full correlation across vertices as expected; explicitly, if f is a Gaussian process on the graph with covariance matrix $D^{1/2} ee^T D^{1/2}$, then $f = v D^{1/2} e$ with v a single Gaussian degree of freedom.

We next consider how random walk kernels on graphs approach the fully correlated case, and show that even for 'simple' graphs the convergence to this limit is non-trivial. Before we do so, we note an additional peculiarity of random walk kernels compared to their Euclidean counterparts: in addition to the maximum range lengthscale p/a discussed so far, they have a diffusive lengthscale $\sigma = (2p/a)^{1/2}$, which is suggested for large p and a by the lengthscale of the corresponding diffusion kernel (1). This diffusive lengthscale will appear in our analysis of learning curves in the large p-limit Section 3.4.1.

2.1 The d-Regular Tree: A Concrete Example

To begin our discussion of the dependence of the random walk kernel on the lengthscale p/a, we first look at how this kernel behaves on a d-regular graph sampled uniformly from the set of all d-regular graphs. Here d-regular means that all vertices have degree $d_i = d$. For a large enough number of vertices V, typical cycles in such a d-regular graph are also large, of length $O(\log V)$, and can be neglected for calculation of the kernel when $V \to \infty$. We therefore begin by assuming the graph is an infinite tree, and assess later how the cycles that do exist on random d-regular graphs cause departures from this picture.

A d-regular tree is a graph where each vertex has degree d with *no cycles*; it is unique up to permutations of the vertices. Since all vertices on the tree are equivalent, the random walk kernel C_{ij} can only depend on the distance between vertices i and j, that is, the smallest number of steps on the graph required to get from one vertex to the other. Denoting the value of a p-step lazy random walk kernel for vertices a distance l apart by $C_{l,p}$, we can determine these values by recursion over

p as follows:

$$C_{l,p=0} = \delta_{l,0}, \qquad \gamma_{p+1}C_{0,p+1} = \left(1 - \frac{1}{a}\right)C_{0,p} + \frac{1}{a}C_{1,p},$$

$$\gamma_{p+1}C_{l,p+1} = \frac{1}{ad}C_{l-1,p} + \left(1 - \frac{1}{a}\right)C_{l,p} + \frac{d-1}{ad}C_{l+1,p} \quad l \geq 1. \tag{4}$$

Here γ_p is chosen to achieve the desired normalisation of the prior variance for every p. We will normalise so that $C_{0,p} = 1$.

Figure 1 (left) shows the results obtained by iterating Equation (4) numerically for a 3-regular tree with $a = 2$. As expected the kernel becomes longer-ranged initially as p is increased, but seems to approach a non-trivial limiting form. This can be calculated analytically and is given by (see Appendix A)

$$C_{l,p\to\infty} = \left[1 + \frac{l(d-2)}{d}\right]\frac{1}{(d-1)^{l/2}}. \tag{5}$$

Equation (5) can be derived by taking the $\sigma^2 \to \infty$ limit of the integral expression for the diffusion kernel from Chung and Yau (1999) whilst preserving normalisation of the kernel (see Appendix A.1 for further details). Alternatively the result (5) can be obtained by rewriting the random walk in terms of shells, that is, grouping vertices according to distance l from a chosen central vertex. The number of vertices in the l-th shell, or shell volume, is $v_l = d(d-1)^{l-1}$ for $l \geq 1$ and $v_0 = 1$. Introducing $R_{l,p} = C_{l,p}\sqrt{v_l}$, Equation (4) can be written in the form

$$R_{l,p=0} = \delta_{l,0}, \qquad \gamma_{p+1}R_{0,p+1} = \left(1 - \frac{1}{a}\right)R_{0,p} + \frac{1}{a\sqrt{d}}R_{1,p},$$

$$\gamma_{p+1}R_{l,p+1} = \frac{\sqrt{d-1}}{ad}R_{l-1,p} + \left(1 - \frac{1}{a}\right)R_{l,p} + \frac{\sqrt{d-1}}{ad}R_{l+1,p} \quad l \geq 1. \tag{6}$$

This is just the un-normalised diffusion equation for a biased random walk on a one dimensional lattice with a reflective boundary at 0. This has been solved in Monthus and Texier (1996), and mapping this solution back to $C_{l,p}$ gives (5) (see Appendix A.2 for further details).

To summarise thus far, the analysis on a d-regular tree shows that, for large p, the random walk kernel does not approach the expected fully correlated limit: because all vertices have the same degree this limit would correspond to $C_{l,p\to\infty} = 1$. On the other hand, on a d-regular graph with any finite number V of vertices, the fully correlated limit must necessarily be approached as $p \to \infty$. As a large regular graph is locally treelike, the difference must arise from the existence of long cycles in a regular graph.

To estimate when the existence of cycles will start to affect the kernel, consider first a d-regular tree truncated at depth l. This will have $V = 1 + \sum_{i=1}^{l} d(d-1)^{i-1} = O(d(d-1)^{l-1})$ vertices. On a d-regular graph with the same number of vertices, we therefore expect to encounter cycles after a number of steps, taken along the graph, of order l. In the random walk kernel the typical number of steps is p/a, so effects of cycles should appear once p/a becomes larger than

$$\frac{p}{a} \approx \frac{\log(V)}{\log(d-1)}. \tag{7}$$

Figure 1 (right) shows a comparison between $C_{1,p}$ as calculated from Equation (4) for a 3-regular tree and its analogue on random 3-regular graphs of finite size, which we call $K_{1,p}$. We define this
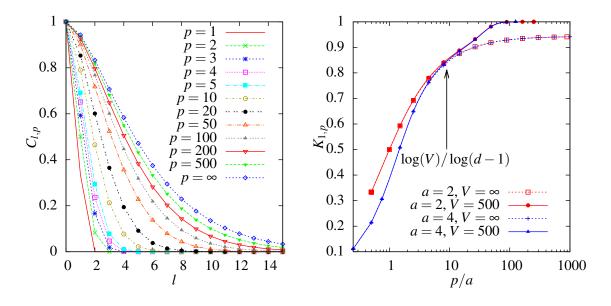
Figure 1: (Left) Random walk kernel $C_{l,p}$ on a 3-regular tree plotted against distance l for increasing number of steps p and $a = 2$. (Right) Comparison between numerical results for the average nearest neighbour kernel $K_{1,p}$ on random 3-regular graphs with the result $C_{1,p}$ on a 3-regular tree, calculated numerically by iteration of (4).

analogue as the average of $C_{ij}/\sqrt{C_{ii}C_{jj}}$ over all pairs of neighbouring vertices on a fixed graph, averaged further over a number of randomly generated regular graphs. The square root accounts for the fact that local kernel values C_{ii} can vary slightly on a regular graph because of cycles, while they are the same for all vertices of a regular tree. Looking at Figure 1 (right) one sees that, as expected from the arguments above, the nearest neighbour kernel value for the 3-regular graph, $K_{1,p}$, coincides with its analogue $C_{1,p}$ on the 3-regular tree for small p. When p/a crosses the threshold (7), cycles in the regular graph become important and the two curves separate. For larger p, the kernel value for neighbouring vertices approaches the fully correlated limit $K_{1,p} \to 1$ on a regular graph, while on a regular tree one has the non-trivial limit $C_{1,p} \to 2\sqrt{d-1}/d$ from (5).

In conclusion of our analysis of random walk kernels, we have seen that these kernels have an unusual dependence on their lengthscale p/a. In particular, kernel values for vertices a short distance apart can remain significantly below the fully correlated limit, even if p/a is large. That limit is approached only once p/a becomes larger than the graph size-dependent threshold (7), at which point cycles become important. We have focused here on random regular graphs, but the same qualitative behaviour should be observed also on graphs with a non-trivial distribution of vertex degrees d_i.

3. Learning Curves for Gaussian Process Regression

Having reached a better understanding of the random walk kernel we now study its application in machine learning. In particular we focus on the use of the random walk kernel for regression with Gaussian processes. We will begin, for completeness, with an introduction to GPs for regres-

sion. For a more comprehensive discussion of GPs for machine learning we direct the reader to Rasmussen and Williams (2005).

3.1 Gaussian Process Regression: Kernels as Covariance Functions

Gaussian process regression is a Bayesian inference technique that constructs a posterior distribution over a function space, $P(f|\boldsymbol{x},\boldsymbol{y})$, given training input locations $\boldsymbol{x} = (x_1,\ldots,x_N)^{\mathrm{T}}$ and corresponding function value outputs $\boldsymbol{y} = (y_1,\ldots,y_N)^{\mathrm{T}}$. The posterior is constructed from a prior distribution $P(f)$ over the function space and the likelihood $P(\boldsymbol{y}|f,\boldsymbol{x})$ to generate the observed output values from function f by using Bayes' theorem

$$P(f|\boldsymbol{x},\boldsymbol{y}) = \frac{P(\boldsymbol{y}|f,\boldsymbol{x})P(f)}{\int \mathrm{d}f' P(\boldsymbol{y}|f',\boldsymbol{x})P(f')}.$$

In the GP setting the prior is chosen to be a Gaussian process, where any finite number of function values has a joint Gaussian distribution, with a covariance matrix with entries given by a *covariance function* or kernel $C(x,x')$ and with a mean vector with entries given by a *mean function* $\mu(x)$. For simplicity we will focus on zero mean GPs[1] and a Gaussian likelihood, which amounts to assuming that training outputs are corrupted by independent and identically distributed Gaussian noise. Under these assumptions all distributions are Gaussian and can be calculated explicitly. If we assume we are given training data $\{(x_\mu,y_\mu)|\mu=1,\ldots,N\}$ where y_μ is the value of the target or 'teacher' function at input location x_μ, corrupted by additive Gaussian noise with variance σ^2, the posterior distribution is then given by another Gaussian process with mean and covariance functions

$$\bar{f}(x) = \boldsymbol{k}(x)^{\mathrm{T}}\boldsymbol{K}^{-1}\boldsymbol{y}, \tag{8}$$

$$\mathrm{Cov}(x,x') = C(x,x') - \boldsymbol{k}(x)^{\mathrm{T}}\boldsymbol{K}^{-1}\boldsymbol{k}(x'), \tag{9}$$

where $\boldsymbol{k}(x) = (C(x_1,x),\ldots,C(x_N,x))^{\mathrm{T}}$ and $K_{\mu\nu} = C(x_\mu,x_\nu) + \delta_{\mu\nu}\sigma^2$. With the posterior in the form of a Gaussian process, predictions are simple. Assuming a squared loss function, the optimal prediction of the outputs is given by $\bar{f}(x)$ and a measure of uncertainty in the prediction is provided by $\mathrm{Cov}(x,x)^{1/2}$.

Equations (8) and (9) illustrate that, in the setting of GP regression, kernels are used to change the type of function preferred by the Gaussian process prior, and correspondingly the posterior. The kernel can encode prior beliefs about smoothness properties, lengthscale and expected amplitude of the function we are trying to predict. Of particular importance for the discussion below, $C(x,x)$ gives the prior variance of the function f at input x, so that $C(x,x)^{1/2}$ sets the typical function *amplitude* or *scale*.

3.2 Kernel Normalisation

Conventionally one fixes the desired scale of the kernel using a *global normalisation*: denoting the unnormalised kernel by $\hat{C}(x,x')$ one scales $C(x,x') = \hat{C}(x,x')/\kappa$ to achieve a desired average of $C(x,x)$ across input locations x. In Euclidean spaces one typically uses translationally invariant kernels like the squared exponential kernel. For these, $C(x,x)$ is the same for all input locations x and so global normalisation is sufficient to fix a spatially uniform scale for the prior amplitude. In the case of kernels on graphs, on the other hand, the local connectivity structure around each vertex

---

1. In the discussion and analysis that follows, generalisation to non-zero mean GPs is straightforward.

can be different. Since information about correlations 'propagates' only along graph edges, graph kernels are not generally translation invariant. In particular, there can be large variation among the prior variances at different vertices. This is usually undesirable in a probabilistic model, unless one has strong prior knowledge to justify such variation. For the random walk kernel, the local prior variances are the diagonal entries of Equation (3). These are directly related to the probability of return of a lazy random walk on a graph, which depends sensitively on the local graph structure. This dependence is in general non-trivial, and not just expressible through, for example, the degree of the local vertex. It seems difficult to imagine a scenario where such a link between prior variances and local graph structures could be justified by prior knowledge.

To emphasise the issue, Figure 2 shows examples of distributions of local prior variances C_{ii} for random walk kernels globally normalised to an average prior variance of unity.[2] The distributions are peaked around the desired value of unity but contain many 'outliers' from vertices with abnormally low or high prior variance. Figure 2 (left) shows the distribution of C_{ii} for a large single instance of an Erdős-Rényi random graph (Erdős and Rényi, 1959). In such graphs, each edge is present independently of all others with some fixed probability, giving a Poisson distribution of degrees $p_\lambda(d) = \lambda^d \exp(-\lambda)/d!$; for the figure we chose average degree $\lambda = 3$. Figure 2 (right) shows analogous results for a generalised random graph with power law mixing distribution (Britton et al., 2006). Generalised random graphs are an extension of Erdős-Rényi random graphs where different edges are assigned different probabilities of being present. By appropriate choice of these probabilities (Britton et al., 2006), one can generate a degree distribution that is a superposition of Poisson distributions, $p(d) = \int d\lambda \, p_\lambda(d)p(\lambda)$. We have taken a shifted Pareto distribution, $p(\lambda) = \alpha\lambda_m^\alpha/\lambda^{\alpha+1}$ with exponent $\alpha = 2.5$ and lower cutoff $\lambda_m = 2$ for the distribution of the means.

Looking first at Figure 2 (left), we know that large Erdős-Rényi graphs are locally tree-like and hence one might expect that this would lead to relatively uniform local prior variances. As shown in the figure, however, even for such tree-like graphs large variations can exist in the local prior variances. To give some specific examples, the large spike near 0 is caused by single disconnected vertices and the smaller spike at around 6.8 arises from two-vertex (single edge) disconnected subgraphs. Single vertex subgraphs have an atypically small prior variance since, for a single disconnected vertex i, before normalisation $C_{ii} = (1 - a^{-1})^p$ which is the $q = 0$ contribution from Equation (3). Other vertices in the graph will get additional contributions from $q \geq 1$ and so have a larger prior variance. This effect will become more pronounced as p is increased and the binomial weights assign less weight to the $q = 0$ term.

Somewhat surprisingly at first sight, the opposite effect is seen for two-vertex disconnected subgraphs as shown by the spike around $C_{ii} = 6.8$ in Figure 2 (left). For vertices on such subgraphs, $C_{ii} = \sum_{q=0}^{\lfloor p/2 \rfloor} \binom{p}{2q} a^{-2q}(1 - a^{-1})^{p-2q}$, which is an atypically large return probability: after any even number of steps, the walker must always return to its starting vertex. A similar situation would occur on vertices at the centre of a star. This illustrates that local properties of a vertex alone, like its degree, do not constrain the prior variance. In a two-vertex disconnected subgraph both vertices have degree 1. But there will generically be other vertices of degree 1 that are dangling ends of a large connected graph component, and these will not have similarly elevated return probabilities. Thus, local graph structure is intertwined in a complex manner with local prior variance.

The black line in Figure 2 (left) shows theoretical predictions (see Section 4.2) for the prior variance distribution in the large graph limit. There is significant fine structure in the various peaks,

---

2. We use C_{ii} again here, instead of $C(i,i)$ as in our general discussion of GPs; the subscript notation is more intuitive because the covariance function on a graph is just a $V \times V$ matrix.

on which theory and simulations agree well where the latter give reliable statistics. The decay from the mean is roughly exponential (see linear-log plot in inset), emphasizing that the distribution of local prior variances is not only rather broad but can also have large tails.

For the power law random graph, Figure 2 (right), the broad features of the distribution of local prior variances C_{ii} are similar: a peak at the desired value of unity, overlaid by spikes which again come from single and two-vertex disconnected subgraphs. The inset shows that the tail beyond the mean is roughly exponential again, but with a slower decay; this is to be expected since power law graphs exhibit many more different local structures with a significantly larger probability than is the case for Erdős-Rényi graphs. Accordingly, the distribution of the C_{ii} also has a larger standard deviation than for the Erdős-Rényi case. The maximum values of C_{ii} that we see in these two specific graph instances follow the same trend, with $\max_i C_{ii} \approx 40$ for the power law graph and $\max_i C_{ii} \approx 15$ for the Erdős-Rényi graph. Such large values would constitute rather unrealistic prior assumptions about the scaling of the target function at these vertices.

To summarise, Figure 2 shows that after global normalisation a random walk kernel can retain a large spread in the local prior variances, with the latter depending on the graph structure in a complicated manner. We propose that to overcome this one should use a *local normalisation*. For a desired prior variance c this means normalising according to $C_{ij} = c\hat{C}_{ij}/(\kappa_i \kappa_j)^{1/2}$ with local normalisation constants $\kappa_i = \hat{C}_{ii}$; here $\hat{C}_{ij}$ is the unnormalised kernel matrix as before. This guarantees that all vertices have exactly equal prior variance as in the Euclidean case, that is, all vertices have a prior variance of c. No uncontrolled local variation in the scaling of the function prior then remains, and the computational overhead of local over global normalisation is negligible. Graphically, if we were to normalise the kernel to unity according to the local prescription, a plot of prior variances like the one in Figure 2 would be a delta peak centred at 1.

The effect of this normalisation on the behaviour of GP regression is a key question for the remainder of this paper; numerical simulation results are shown in Section 3.3 below, while our theoretical analysis is described in Section 4.

3.3 Predicting the Learning Curve

The performance of non-parametric methods such as GPs can be characterised by studying the *learning curve*,

$$\varepsilon(N) = \left\langle \left\langle \left\langle \left\langle \frac{1}{V} \sum_{i=1}^{V} \left(g_i - \langle f_i \rangle_{f|x,y}\right)^2 \right\rangle_{y|g,x} \right\rangle_{g} \right\rangle_{x} \right\rangle_{\mathcal{G}},$$

defined as the average squared error between the student and teacher's predictions $f = (f_1, \ldots, f_V)^{\mathrm{T}}$ and $g = (g_1, \ldots, g_V)^{\mathrm{T}}$ respectively, averaged over the student's posterior distribution given the data $f|x,y$, the outputs given the teacher $y|g,x$, the teacher functions g, and the input locations x. This gives the average generalisation error as a function of the number of training examples. For simplicity we will assume that the input distribution is uniform across the vertices of the graph.

Because we are analysing GP regression on graphs, after the averages discussed so far the generalisation error will still depend on the structure of the specific graph considered. We therefore include an additional average, over all graphs in a random graph ensemble $\mathcal{G}$. We consider graph ensembles defined by the distribution of degrees d_i: we specify a degree sequence $\{d_1, \ldots, d_V\}$, or, for large V, equivalently a degree distribution $p(d)$, and pick uniformly at random any one of the graphs that has this degree distribution. The actual shape of the degree distribution is left arbitrary,
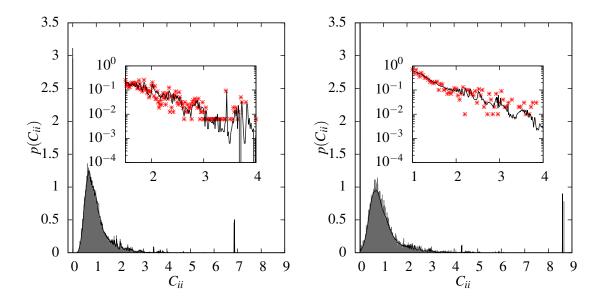
Figure 2: (Left) Grey: histogram of prior variances for the globally normalised random walk kernel with $a = 2$, $p = 10$ on a single instance of an Erdős-Rényi graph with mean degree $\lambda = 3$ and $V = 10000$ vertices. Black: prediction for this distribution in the large graph limit (see Section 4.2). Inset: Linear-log plot of the tail of the distribution. (Right) As (left) but for a power law generalised random graph with exponent 2.5 and cutoff 2.

as long as it has finite mean. Our analysis therefore has broad applicability, including in particular the graph types already mentioned above (d-regular graphs, where $p(d') = \delta_{dd'}$, Erdős-Rényi graphs, power law generalised random graphs).

For this paper, as is typical for learning curve studies, we will assume that teacher and student have the same prior distribution over functions, and likewise that the assumed Gaussian noise of variance σ^2 reflects the actual noise process corrupting the training data. This is known as the *matched case*.[3] Under this assumption the generalisation error becomes the Bayes error, which given that we are considering squared error simplifies to the posterior variance of the student averaged over data sets and graphs (Rasmussen and Williams, 2005). Since we only need the posterior variance we shift f so that the posterior mean is 0; f_i is then just the deviation of the function value at vertex i from the posterior mean. The Bayes error can now be written as

$$\varepsilon(N) = \left\langle \left\langle \left\langle \frac{1}{V} \sum_{i=1}^{V} f_i^2 \right\rangle_{f|x} \right\rangle_x \right\rangle_G . \tag{10}$$

Note that by shifting the posterior distribution to zero mean, we have eliminated the dependence on y in the above equation. That this should be so can also be seen from (9) for the posterior (co-)variance, which only depends on training inputs x but not the corresponding outputs y.

---

3. The case of mismatch has been considered in Malzahn and Opper (2005) for fixed teacher functions, and for prior and noise level mismatch in Sollich (2002); Sollich and Williams (2005).

The averages in Equation (10) are in general difficult to calculate analytically, because the training input locations $\boldsymbol{x}$ enter in a highly nonlinear matter, see (9); only for very specific situations can exact results be obtained (Malzahn and Opper, 2005; Rasmussen and Williams, 2005). Approximate learning curve predictions have been derived, for Euclidean input spaces, with some degree of success (Sollich, 1999a,b; Opper and Vivarelli, 1999; Williams and Vivarelli, 2000; Malzahn and Opper, 2003; Sollich, 2002; Sollich and Halees, 2002; Sollich and Williams, 2005). We will show that in the case of GP regression for functions defined on graphs, learning curves can be predicted exactly in the limit of large random graphs. This prediction is broadly applicable because the degree distribution that specifies the graph ensemble is essentially arbitrary.

It is instructive to begin our analysis by extending a previous approximation seen in Sollich (1999a) and Malzahn and Opper (2005) to our discrete graph case. In so doing we will see explicitly how one may improve this approximation to fully exploit the structure of random graphs, using belief propagation or equivalently the *cavity method* (Mézard and Parisi, 2003). We will sketch the derivation of the existing approximation following the method of Malzahn and Opper (2005); the result given by Sollich (1999a) is included in this as a somewhat more restricted approximation. Both the approximate treatment and our cavity method take a statistical mechanics approach, so we begin by rewriting Equation (10) in terms of a *generating* or *partition function Z*

$$\varepsilon(N) = \left\langle \frac{1}{V} \sum_i \int d\boldsymbol{f} P(\boldsymbol{f}|\boldsymbol{x}) f_i^2 \right\rangle_{\boldsymbol{x},\mathcal{G}} = -\lim_{\lambda \to 0} \frac{2}{V} \frac{\partial}{\partial \lambda} \langle \log(Z) \rangle_{\boldsymbol{x},\mathcal{G}}, \tag{11}$$

with

$$Z = \int d\boldsymbol{f} \exp\left(-\frac{1}{2} \boldsymbol{f}^{\mathrm{T}} C^{-1} \boldsymbol{f} - \frac{1}{2\sigma^2} \sum_{\mu=1}^{N} f_{x_\mu}^2 - \frac{\lambda}{2} \sum_i f_i^2 \right).$$

In this representation the inputs $\boldsymbol{x}$ only enter Z through the sum over μ. We introduce γ_i to count the number of examples at vertex i so that Z becomes

$$Z = \int d\boldsymbol{f} \exp\left(-\frac{1}{2} \boldsymbol{f}^{\mathrm{T}} C^{-1} \boldsymbol{f} - \frac{1}{2} \boldsymbol{f}^{\mathrm{T}} \mathrm{diag}\left(\frac{\gamma_i}{\sigma^2} + \lambda \right) \boldsymbol{f} \right). \tag{12}$$

The average in Equation (11) of the logarithm of this partition function can still not be carried out in closed form. The approximation given by Malzahn and Opper (2005) and our present cavity approach diverge at this point. Section 3.4 discusses the existing approximation for the learning curve, applied to the case of regression on a graph. Section 4 then improves on this using the cavity method to fully exploit the graph structure.

3.4 Kernel Eigenvalue Approximation

The approach of Malzahn and Opper (2005) is to average $\log(Z)$ from (12) using the replica trick (Mézard et al., 1987). One writes $\langle \log Z \rangle_{\boldsymbol{x}} = \lim_{n \to 0} \frac{1}{n} \log \langle Z^n \rangle_{\boldsymbol{x}}$, performing the average $\langle Z^n \rangle_{\boldsymbol{x}}$ for integer n and assuming that a continuation to $n \to 0$ is possible. The required n-th power of Equation (12) is given by

$$\langle Z^n \rangle_{\boldsymbol{x}} = \int \prod_{a=1}^{n} d\boldsymbol{f}^a \left\langle \exp\left(-\frac{1}{2} \sum_a (\boldsymbol{f}^a)^{\mathrm{T}} C^{-1} \boldsymbol{f}^a - \frac{1}{2\sigma^2} \sum_{i,a} \gamma_i (f_i^a)^2 - \frac{\lambda}{2} \sum_{i,a} (f_i^a)^2 \right) \right\rangle_{\boldsymbol{x}},$$

where the replica index a runs from 1 to n. Assuming as before that examples are generated independently and uniformly from $\mathcal{V}$, the data set average over $\boldsymbol{x}$ will, for large V, become equivalent to independent Poisson averages over γ_i with mean $\nu = N/V$. Explicitly performing these averages gives

$$\langle Z^n \rangle_{\boldsymbol{x}} = \int \prod_{a=1}^{n} d\boldsymbol{f}^a \exp\left(-\frac{1}{2}\sum_a (\boldsymbol{f}^a)^{\mathrm{T}} C^{-1} \boldsymbol{f}^a + \nu \sum_i \left(e^{-\Sigma_a (f_i^a)^2/2\sigma^2} - 1 \right) - \frac{\lambda}{2}\sum_{i,a} (f_i^a)^2 \right). \quad (13)$$

In order to evaluate (13) one has to find a way to deal with the exponential term in the exponent. Malzahn and Opper (2005) do this using a variational approximation for the distribution of the $\boldsymbol{f}^a$, of Gaussian form. Eventually this leads to the following eigenvalue learning curve approximation (see also Sollich, 1999a):

$$\varepsilon(N) = g\left(\frac{N}{\varepsilon(N) + \sigma^2} \right), \qquad g(h) = \sum_{\alpha=1}^{V} \left(\lambda_\alpha^{-1} + h \right)^{-1}. \quad (14)$$

The eigenvalues λ_α of the kernel are defined here from the eigenvalue equation[4] $(1/V)\sum_j C_{ij}\phi_j = \lambda\phi_i$. The Gaussian variational approach is evidently justified for large σ^2, where a Taylor expansion of the exponential term in (13) can be truncated after the quadratic term. For small noise levels, on the other hand, the Gaussian variational approach will in general not capture all the details of the fluctuations in the numbers of examples γ_i. This issue is expected to be most prominent for values of ν of order unity, where fluctuations in the number of examples are most relevant because some vertices will not have seen examples locally or nearby and will have posterior variance close to the prior variance, whereas those vertices with examples will have small posterior variance, of order σ^2. This effect disappears again for large ν, where the $O(\sqrt{\nu})$ fluctuations in the number of examples at each vertex becomes relatively small. Mathematically this can be seen from the term proportional to ν in (13), which for large ν ensures that only values of f_i^a with $\exp(-\sum_a(f_i^a)^2/2\sigma^2)$ close to 1 will contribute. A quadratic approximation is then justified even if σ^2 is not large.

Learning curve predictions from Equation (14) using numerically computed eigenvalues for the globally normalised random walk kernel are shown in Figure 3 as dotted lines for random regular (left), Erdős-Rényi (centre) and power law generalised random graphs (right). The predictions are compared to numerically simulated learning curves shown as solid lines, for a range of noise levels. Consistent with the discussion above, the predictions of the eigenvalue approximation are accurate where the Gaussian variational approach is justified, that is, for small and large ν. Figure 3 also shows that the accuracy of the approximation improves as the noise level σ^2 becomes larger, again as expected by the nature of the Gaussian approximation.

3.4.1 LEARNING CURVES FOR LARGE p

Before moving on to the more accurate cavity prediction of the learning curves, we now look at how the learning curves for GP regression on graphs depend on the kernel lengthscale p/a. We focus for this discussion on random regular graphs, where the distinction between global and local normalisation is not important. In Section 2.1, we saw that on a large regular graph the random walk kernel approaches a non-trivial limiting form for large p, as long as one stays below the threshold

---

4. Here and below we consider the case of a uniform distribution of inputs across vertices, though the results can be generalised to the non-uniform case.
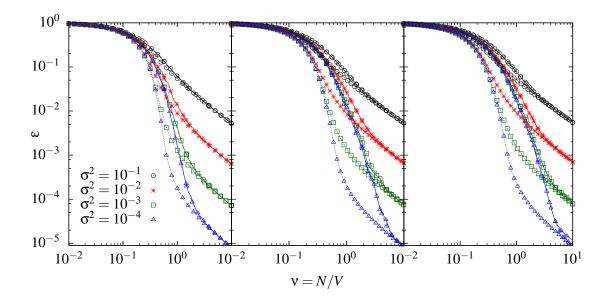
Figure 3: (Left) Learning curves for GP regression with globally normalised kernels with $p = 10$, $a = 2$ on 3-regular random graphs for a range of noise levels σ^2. Dotted lines: eigenvalue predictions (see Section 3.4), solid lines: numerically simulated learning curves for graphs of size $V = 500$, dashed lines: cavity predictions (see Section 4.1); note these are mostly visually indistinguishable from the simulation results. (Centre) As (left) for Erdős-Rényi random graphs with mean degree 3. (Right) As (left) for power law generalised random graphs with exponent 2.5 and cutoff 2.

(7) for p where cycles become important. One might be tempted to conclude from this that also the learning curves have a limiting form for large p. This is too naive however, as one can see by considering, for example, the effect of the first example on the Bayes error. If the example is at vertex i, the posterior variance at vertex j is, from (9), $C_{jj} - C_{ij}^2/(C_{ii} + \sigma^2)$. As the prior variances C_{jj} are all equal, to unity for our chosen normalisation, this is $1 - C_{ij}^2/(1 + \sigma^2)$. The reduction in the Bayes error is therefore $\varepsilon(0) - \varepsilon(1) = (1/V)\sum_j C_{ij}^2/(1 + \sigma^2)$. As long as cycles are unimportant this is independent of the location of the example vertex i, and in the notation of Section 2.1 can be written as

$$\varepsilon(0) - \varepsilon(1) = \frac{1}{1 + \sigma^2} \sum_{l=0}^{p} v_l C_{l,p}^2, \tag{15}$$

where v_l is, as before, the number of vertices a distance l away from vertex i, that is, $v_0 = 1$, $v_l = d(d-1)^{l-1}$ for $l \geq 1$. To evaluate (15) for large p, one cannot directly plug in the limiting kernel form (5): the 'shell volume' v_l just balances the l-dependence of the factor $(d-1)^{-l/2}$ from $C_{l,p}$, so that one gets contributions from all distances l, proportional to l^2 for large l. Naively summing up to $l = p$ would give an initial decrease of the Bayes error growing as p^3. This is not correct; the reason is that while $C_{l,p}$ approaches the large p-limit (5) for any fixed l, it does so more and more slowly as l increases. A more detailed analysis, sketched in Appendix A.2, shows that

for large l and p, $C_{l,p}$ is proportional to the large p-limit $l(d-1)^{-l/2}$ up to a characteristic cutoff distance l of order $p^{1/2}$, and decays quickly beyond this. Summing in (15) the contributions of order l^2 up to this distance predicts finally that the initial error decay should scale, non-trivially, as $p^{3/2}$.

We next show that this large p-scaling with $p^{3/2}$ is also predicted, for the entire learning curve, by the eigenvalue approximation (14). As before we consider d-regular random graphs. The required spectrum of kernel eigenvalues λ_α becomes identical, for large V, to that on a d-regular tree (McKay, 1981). Explicitly, if λ_α^L are the eigenvalues of the normalised graph Laplacian on a tree, then the kernel eigenvalues are $\lambda_\alpha = \kappa^{-1} V^{-1} (1 - \lambda_\alpha^L / a)^p$. Here the factor V^{-1} comes from the same factor in the kernel eigenvalue definition after (14), and κ is the overall normalisation constant which enforces $\sum_\alpha \lambda_\alpha = V^{-1} \sum_j C_{jj} = 1$. The spectrum of the tree Laplacian is known (see McKay, 1981; Chung, 1996) and is given by

$$\rho(\lambda^L) = \begin{cases} \dfrac{\sqrt{\frac{4(d-1)}{d^2} - (\lambda^L - 1)^2}}{(2\pi/d)\lambda^L(2-\lambda^L)} & \lambda_- \leq \lambda \leq \lambda_+, \\ 0 & \text{otherwise,} \end{cases}$$

where $\lambda_\pm = 1 \pm \frac{2}{d}(d-1)^{1/2}$. (There are also two isolated eigenvalues at 0 and 2, which do not contribute for large V.)

We can now write down the function g from (14), converting the sum over kernel eigenvalues to V times an integral over Laplacian eigenvalues for large V. Dropping the L superscript, the result is

$$g(h) = \int_{\lambda_-}^{\lambda_+} \mathrm{d}\lambda \, \rho(\lambda) [\kappa(1 - \lambda/a)^{-p} + hV^{-1}]^{-1}. \tag{16}$$

The dependence on hV^{-1} here shows that in the approximate learning curve (14), the Bayes error will depend only on $\nu = N/V$ as might have been expected. The condition for the normalisation factor κ becomes simply $g(0) = 1$, or $\kappa^{-1} = \int \mathrm{d}\lambda \, \rho(\lambda)(1 - \lambda/a)^p$.

So far we have written down how one would evaluate the eigenvalue approximation to the learning curve on large d-regular random graphs, for arbitrary kernel parameters p and a. Now we want to consider the large p-limit. We show that there is then a *master curve* for the Bayes error against $\nu p^{3/2}$. This is entirely consistent with the $p^{3/2}$ scaling found above for the initial error decay. The intuition for the large p analysis is that the factor $(1 - \lambda/a)^p$ decays quickly as the Laplacian eigenvalue λ increases beyond λ_-, so that only values of λ near λ_- contribute. One can then approximate

$$\left(1 - \frac{\lambda}{a}\right)^p \approx \left(1 - \frac{\lambda_-}{a}\right)^p \exp\left(-\frac{p(\lambda - \lambda_-)}{a - \lambda_-}\right).$$

Similarly one can replace $\rho(\lambda)$ by its leading square root behaviour near λ_-,

$$\rho(\lambda) = (\lambda - \lambda_-)^{1/2} \frac{(d-1)^{1/4} d^{5/2}}{\pi(d-2)^2}.$$

Substituting these approximations into (16) and introducing the rescaled integration variable $y = p(\lambda - \lambda_-)/(a - \lambda_-)$ gives

$$g(h) = r\kappa^{-1}(1 - \lambda_-/a)^p \left(\frac{a - \lambda_-}{p}\right)^{3/2} F(h\kappa^{-1} V^{-1}(1 - \lambda_-/a)^p),$$
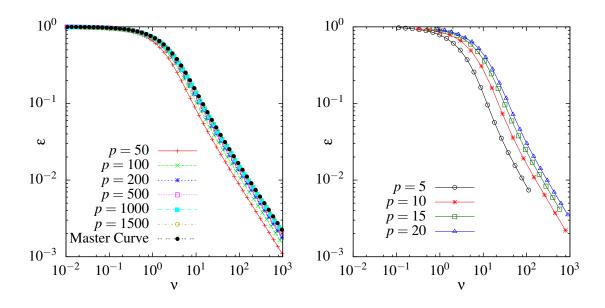
Figure 4: (Left) Eigenvalue approximation for learning curves on a random 3-regular graph, using a random walk kernel with $a = 2$, $\sigma^2 = 0.1$ and increasing values of p as shown. Plotting against $\nu p^{3/2}$ shows that for large p these rescaled curves approach the master curve predicted from (17), though this approach is slower in the tail of the curves. (Right) As (left), but for numerically simulated learning curves on graphs of size $V = 500$.

where $r = (d-1)^{1/4}d^{5/2}/(\pi(d-2)^2)$ and $F(z) = \int_0^\infty dy\, y^{1/2}(\exp(y)+z)^{-1}$. Since $g(0) = 1$, the prefactor must equal $1/F(0) = 2/\sqrt{\pi}$. This fixes the normalisation constant κ, and we can simplify to

$$g(h) = \frac{F(hV^{-1}c^{-1})}{F(0)}, \qquad c = rF(0)\left(\frac{a-\lambda_-}{p}\right)^{3/2}.$$

The learning curves for large p are then predicted from (14) by solving

$$\varepsilon = F(\nu c^{-1}/(\varepsilon+\sigma^2))/F(0), \tag{17}$$

and depend clearly only on the combination νc^{-1}. Because c is proportional to $p^{-3/2}$, this shows that learning curves for different p should collapse onto a master curve when plotted against $\nu p^{3/2}$.

A plot of the scaling of the eigenvalue learning curve approximations onto the master curve is shown in Figure 4 (left). As can be seen, large values of p are required in order to get a good collapse in the tail of the learning curve prediction, whereas in the initial part the $p^{3/2}$ scaling is accurate already for relatively small p.

Finally, Figure 4 (right) shows that the predicted $p^{3/2}$-scaling of the learning curves is present not only within the eigenvalue approximation, but also in the actual learning curves. Figure 4 (right) displays numerically simulated learning curves for $p = 5, 10, 15$ and 20, against the rescaled number of examples $\nu p^{3/2}$ as before. Even for these comparatively small values of p one sees that the rescaled learning curves approach a master curve.

4. Exact Learning Curves: Cavity Method

So far we have discussed the eigenvalue approximation of GP learning curves, and how it deviates from numerically exact simulated learning curves. As discussed in Section 3.4, the deficiencies of the eigenvalue approximation can be traced back to the fact that the fluctuations in the number of training examples seen at each vertex of the graph cannot be accounted for in detail. If in the average over data sets these fluctuations could be treated exactly, one would hope to obtain exact, or at least very accurate, learning curve predictions. In this section we show that this is indeed possible in the case of a random walk kernel, for both global and local normalisations. We derive our prediction using belief propagation or, equivalently, the cavity method (Mézard and Parisi, 2003). The approach relies on the fact that the local structure of the graph on which we are learning is tree-like. This local tree-like structure always occurs in large random graphs sampled uniformly from an ensemble specified by an arbitrary but fixed degree distribution, which is the scenario we consider here. We will see that already for moderate graph sizes of $V = 500$, our predictions are nearly indistinguishable from numerical simulations.

In order to apply the cavity method to the problem of predicting learning curves we must first rewrite the partition function (12) in the form of a graphical model. This means that the function being integrated over to obtain Z must consist of factors relating only to individual vertices, or to pairs of neighbouring vertices. The inverse of the covariance matrix in (12) creates factors linking vertices at arbitrary distances along the graph, and so must be eliminated before the cavity method can be applied. We begin by assuming a general form for the normalisation of $\hat{C}$ that encompasses both local and global normalisation and set $C = \mathcal{K}^{-1/2}[(1-a^{-1})I + a^{-1}D^{-1/2}AD^{-1/2}]^p \mathcal{K}^{-1/2}$ with $\mathcal{K}_{ij} = \kappa_i \delta_{ij}$. To eliminate interactions across the entire graph we first Fourier transform the prior term $\exp(-\frac{1}{2}f^T C^{-1} f)$ in (12), introduce Fourier variables h, and then integrate out the remaining terms with respect to f to give

$$Z \propto \prod_i \left(\frac{\gamma_i}{\sigma^2} + \lambda\right)^{-1/2} \int dh \exp\left(-\frac{1}{2}h^T C h - \frac{1}{2}h^T \text{diag}\left(\left(\frac{\gamma_i}{\sigma^2} + \lambda\right)^{-1/2}\right) h\right).$$

The coupling between different vertices in (4) is now through C so still links vertices up to distance p. To reduce these remaining interactions to ones among nearest neighbours only, one exploits the binomial expansion of the random walk kernel given in (3). Defining p additional variables at each vertex as $h^q = \mathcal{K}^{1/2}(D^{-1/2}AD^{-1/2})^q \mathcal{K}^{-1/2}h$, $q = 1, \ldots, p$, and abbreviating $c_q = \binom{p}{q}(1 - a^{-1})^{p-q}(a^{-1})^q$, the interaction term $h^T C h$ turns into a local term $\sum_{q=0}^{p} c_q (h^0)^T \mathcal{K}^{-1} h^q$. (Here we have, for the sake of uniformity, written h^0 instead of h.) Of course the interactions have only been 'hidden' in the h^q, but the key point is that the definition of these additional variables can be enforced recursively, via $h^q = \mathcal{K}^{1/2}D^{-1/2}AD^{-1/2}\mathcal{K}^{-1/2}h^{q-1}$. We represent this definition via a Dirac delta function (for each $q = 1, \ldots, p$) and then Fourier transform the latter, with conjugate variables $\hat{h}^q$, to get

$$Z \propto \prod_i \left(\frac{\gamma_i}{\sigma^2} + \lambda\right)^{-1/2} \int \prod_{q=0}^{p} dh^q \prod_{q=1}^{p} d\hat{h}^q \exp\left(-\frac{1}{2}(h^0)^T \text{diag}\left(\left(\frac{\gamma_i}{\sigma^2} + \lambda\right)^{-1}\right) h^0\right.$$

$$\left. -\frac{1}{2}\sum_{q=0}^{p} c_q (h^0)^T \mathcal{K}^{-1} h^q + i\sum_{q=1}^{p} (\hat{h}^q)^T \left(h^q - \mathcal{K}^{1/2}D^{-1/2}AD^{-1/2}\mathcal{K}^{-1/2}h^{q-1}\right)\right). \quad (18)$$

Because the graph adjacency matrix $\boldsymbol{A}$ now appears at most linearly in the exponent, all interactions are between nearest neighbours only. We have thus expressed our Z as the partition function of a (complex-valued) graphical model.

4.1 Global Normalisation

We can now apply belief propagation to the calculation of marginals for the above graphical model. We focus first on the simpler case of a globally normalised kernel where $\kappa_i = \kappa$ for all i. Rescaling each h_i^q to $d_i^{1/2}\kappa^{1/2}h_i^q$ and $\hat{h}_i^q$ to $d_i^{1/2}\hat{h}_i^q/\kappa^{1/2}$ we are left with

$$
Z \propto \prod_i \left(\frac{\gamma_i}{\sigma^2}+\lambda\right)^{-1/2} \int \prod_{q=0}^p \mathrm{d}h^q \prod_{q=1}^p \mathrm{d}\hat{h}^q \prod_i \exp\left(-\frac{1}{2}\sum_{q=0}^p c_q h_i^0 h_i^q d_i - \frac{1}{2}\frac{(h_i^0)^2 \kappa d_i}{\gamma_i/\sigma^2+\lambda} + \mathrm{i}\sum_{q=1}^p d_i \hat{h}_i^q h_i^q\right)
$$
$$
\prod_{(i,j)} \exp\left(-\mathrm{i}\sum_{q=1}^p \left(\hat{h}_i^q h_j^{q-1}+\hat{h}_j^q h_i^{q-1}\right)\right), \quad (19)
$$

where the interaction terms coming from the adjacency matrix, $\boldsymbol{A}$, have been written explicitly as a product over distinct graph edges (i,j).

To see how the Bayes error (10) can be obtained from this partition function, we differentiate $\log(Z)$ with respect to λ as prescribed by (11) to get

$$
\varepsilon(\nu) = \lim_{\lambda\to 0}\frac{1}{V}\sum_i \frac{1}{\gamma_i/\sigma^2+\lambda}\left(1 - \frac{d_i\kappa\langle(h_i^0)^2\rangle}{\gamma_i/\sigma^2+\lambda}\right). \quad (20)
$$

In order to calculate the Bayes error we therefore require specifically the marginal distributions of h_i^0. These can be calculated using the cavity method: for a large random graph with arbitrary fixed degree sequence the graph is locally tree-like, so that if vertex i were eliminated the corresponding subgraphs (locally trees) rooted at the neighbours $j \in \mathcal{N}(i)$ of i would become approximately independent. The resulting cavity marginals created by removing i, which we denote $P_j^{(i)}(\boldsymbol{h}_j,\hat{\boldsymbol{h}}_j|\boldsymbol{x})$, can then be calculated iteratively within these subgraphs using the update equations

$$
P_j^{(i)}(\boldsymbol{h}_j,\hat{\boldsymbol{h}}_j|\boldsymbol{x}) \propto \exp\left(-\frac{1}{2}\sum_{q=0}^p c_q d_j h_j^0 h_j^q - \frac{1}{2}\frac{d_j\kappa(h_j^0)^2}{\gamma_j/\sigma^2+\lambda} + \mathrm{i}\sum_{q=1}^p d_j \hat{h}_j^q h_j^q\right)
$$
$$
\int \prod_{k\in\mathcal{N}(j)\setminus i} \mathrm{d}\boldsymbol{h}_k \mathrm{d}\hat{\boldsymbol{h}}_k \exp\left(-\mathrm{i}\sum_{q=1}^p (\hat{h}_j^q h_k^{q-1}+\hat{h}_k^q h_j^{q-1})\right) P_k^{(j)}(\boldsymbol{h}_k,\hat{\boldsymbol{h}}_k|\boldsymbol{x}). \quad (21)
$$

where $\boldsymbol{h}_j = (h_j^0,\ldots,h_j^p)^{\mathrm{T}}$ and $\hat{\boldsymbol{h}}_j = (\hat{h}_j^1,\ldots,\hat{h}_j^p)^{\mathrm{T}}$. In terms of the sum-product formulation of belief propagation, the cavity marginal on the left is the message that vertex j sends to the factor in Z for edge (i,j) (Bishop, 2007).

One sees that the cavity update Equations (21) are solved self-consistently by complex-valued Gaussian distributions with mean zero and covariance matrices $\boldsymbol{V}_j^{(i)}$. This Gaussian character of the solution was of course to be expected because in (19) we have a Gaussian graphical model. By performing the Gaussian integrals in the cavity update equations explicitly, one finds for the corresponding updates of the covariance matrices the rather simple form

$$
\boldsymbol{V}_j^{(i)} = \left(\boldsymbol{O}_j - \sum_{k\in\mathcal{N}(j)\setminus i} \boldsymbol{X}\boldsymbol{V}_k^{(j)}\boldsymbol{X}\right)^{-1}, \quad (22)
$$

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

$$\boldsymbol{O}_j = d_j \left(\begin{array}{ccccccc} c_0 + \frac{\kappa}{\gamma_j/\sigma^2 + \lambda} & \frac{c_1}{2} & \cdots & \frac{c_p}{2} & 0 & \cdots & 0 \\ \frac{c_1}{2} & & & & -i & & \\ \vdots & & & & & \ddots & \\ \frac{c_p}{2} & & & & & & -i \\ \hline 0 & -i & & & & & \\ \vdots & & \ddots & & & \boldsymbol{0}_{p,p} & \\ 0 & & & -i & & & \end{array} \right), \quad \boldsymbol{X} = \left(\begin{array}{cc|ccc} & & i & & \\ & \boldsymbol{0}_{p+1,p+1} & & \ddots & \\ & & & & i \\ & & 0 & \cdots & 0 \\ \hline i & & & & \\ & \ddots & & \vdots & \boldsymbol{0}_{p,p} \\ & & i & 0 & \end{array} \right). \quad (23)$$

At first glance (22) becomes singular for $\gamma_j = 0$; however this is easily avoided. We introduce $\boldsymbol{O}_j - \sum_{k=1}^{d-1} \boldsymbol{X} \boldsymbol{V}_k^{(j)} \boldsymbol{X} = \boldsymbol{M}_j + [d_j \kappa / (\gamma_j / \sigma^2 + \lambda)] \boldsymbol{e}_0 \boldsymbol{e}_0^T$ with $\boldsymbol{e}_0^T = (1, 0, \ldots, 0)$ so that $\boldsymbol{M}_j$ contains all the non-singular terms. We may then apply the Woodbury identity (Hager, 1989) to write the matrix inverse in a form where the $\lambda \to 0$ limit can be taken without difficulties:

$$\left(\boldsymbol{O}_j - \sum_{k=1}^{d-1} \boldsymbol{X} \boldsymbol{V}_k^{(j)} \boldsymbol{X} \right)^{-1} = \boldsymbol{M}_j^{-1} - \frac{\boldsymbol{M}_j^{-1} \boldsymbol{e}_0 \boldsymbol{e}_0^T \boldsymbol{M}_j^{-1}}{(\gamma_j / \sigma^2 + \lambda)/(d_j \kappa) + \boldsymbol{e}_0^T \boldsymbol{M}_j^{-1} \boldsymbol{e}_0}.$$

In our derivation so far we have assumed a fixed graph, we therefore need to translate these equations to the setting we ultimately want to study, that is, an ensemble of large random graphs. This ensemble is characterised by the distribution $p(d)$ of the degrees d_i, so that every graph that has the desired degree distribution is assigned equal probability. Instead of individual cavity covariance matrices $\boldsymbol{V}_j^{(i)}$, one must then 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 vertex j will have degree d_j is then $p(d_j) d_j / \bar{d}$, because such a vertex has d_j 'chances' of being picked. (The normalisation factor is the average degree $\bar{d} = \sum_i p(d_i) d_i$.) Using again the locally treelike structure, the incoming (to vertex j) cavity covariances $\boldsymbol{V}_k^{(j)}$ will be independent and identically distributed samples from $W(\boldsymbol{V})$. Thus a fixed point of the cavity update equations corresponds to a fixed point of an update equation for $W(\boldsymbol{V})$:

$$W(\boldsymbol{V}) = \sum_d \frac{p(d)d}{\bar{d}} \left\langle \int \prod_{k=1}^{d-1} d\boldsymbol{V}_k W(\boldsymbol{V}_k) \, \delta \left(\boldsymbol{V} - \left(\boldsymbol{O} - \sum_{k=1}^{d-1} \boldsymbol{X} \boldsymbol{V}_k \boldsymbol{X} \right)^{-1} \right) \right\rangle_\gamma. \quad (24)$$

Since the vertex label is now arbitrary, we have omitted the index j. The average in (24) is over the distribution of the number of examples $\gamma \equiv \gamma_j$ at vertex j. As before we assume for simplicity that examples are drawn with uniform input probability across all vertices, so that the distribution of γ is simply $\gamma \sim \text{Poisson}(\nu)$ in the limit of large N and V at fixed $\nu = N/V$.

In general Equation (24)—which can also be formally derived using the replica approach (see Urry and Sollich, 2012)—cannot be solved analytically, but we can tackle it numerically using population dynamics (Mézard and Parisi, 2001). This is an iterative technique where one creates a population of covariance matrices and for each iteration updates a random element of the population according to the delta function in (24). The update is calculated by sampling from the degree distribution $p(d)$ of local degrees, the Poisson distribution of the local number of examples ν and

from the distribution $W(V_k)$ of 'incoming' covariance matrices, the latter being approximated by uniform sampling from the current population.

Once we have $W(V)$, the Bayes error can be found from the graph ensemble version of Equation (11). This is obtained by inserting the explicit expression for $\langle (h_i^0)^2 \rangle$ in terms of the cavity marginals of the neighbouring vertices, and replacing the average over vertices with an average over degrees d:

$$\varepsilon(\nu) = \lim_{\lambda \to 0} \sum_d p(d) \left\langle \frac{1}{\gamma/\sigma^2 + \lambda} \left(1 - \frac{d\kappa}{\gamma/\sigma^2 + \lambda} \int \prod_{k=1}^d dV_k W(V_k) \, (O - \sum_{k=1}^d X V_k X)_{00}^{-1} \right) \right\rangle_\gamma . \quad (25)$$

The number of examples at the vertex γ is once more to be averaged over $\gamma \sim \text{Poisson}(\nu)$. The subscript '00' indicates the top left element of the matrix, which determines the variance of h^0.

To be able to use Equation (25), we again need to rewrite it into a form that remains explicitly non-singular when $\gamma = 0$ and $\lambda \to 0$. We separate the γ-dependence of the matrix inverse again and write, in slightly modified notation as appropriate for the graph ensemble case, $O - \sum_{k=1}^d X V_k X = M_d + [d\kappa/(\gamma/\sigma^2 + \lambda)] e_0 e_0^T$, where $e_0^T = (1, 0, \ldots, 0)$. The 00 element of the matrix inverse appearing above can then be expressed using the Woodbury formula (Hager, 1989) as

$$e_0^T \left(O - \sum_{k=1}^d X V_k X \right)^{-1} e_0 = e_0^T M_d^{-1} e_0 - \frac{e_0^T M_d^{-1} e_0 e_0^T M_d^{-1} e_0}{(\gamma/\sigma^2 + \lambda)/(d\kappa) + e_0^T M_d^{-1} e_0}.$$

The $\lambda \to 0$ limit can now be taken, with the result

$$\varepsilon(\nu) = \left\langle \sum_d p(d) \int \prod_{k=1}^d dV_k W(V_k) \frac{1}{\gamma/\sigma^2 + d\kappa (M_d^{-1})_{00}} \right\rangle_\gamma . \quad (26)$$

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

The self-consistency Equation (24) for $W(V)$ and the expression (26) for the resulting Bayes error allow us to predict learning curves as a function of the number of examples per vertex, ν, for *arbitrary degree distributions* $p(d)$ of our random graph ensemble. For large graphs the predictions should become exact. It is worth stressing that such exact learning curve predictions have previously only been available in very specific, noise-free, GP regression scenarios, while our result for GP regression on graphs is applicable to a broad range of random graph ensembles, with arbitrary noise levels and kernel parameters.

We note briefly that for graphs with isolated vertices ($d = 0$), one has to be slightly careful: already in the definition of the covariance function (2) one should replace $D \to D + \delta I$ to avoid division by zero, taking $\delta \to 0$ at the end. For $d = 0$ one then finds in the expression (26) that $(M^{-1})_{00} = 1/(c_0 \delta)$, where c_0 is defined before (18). As a consequence, $\kappa(\delta + d)(M^{-1})_{00} = \kappa\delta(M^{-1})_{00} = \kappa/c_0$. This is to be expected since isolated vertices each have a separate Gaussian prior with variance c_0/κ.

Equations (24) and (26) still require the normalisation constant, κ. The simplest way to calculate this is to run the population dynamics once for $\kappa = 1$ and $\nu = 0$, that is, an unnormalised kernel and no training data. The result for ε then just gives the average (over vertices) prior variance. With κ set to this value, one can then run the population dynamics for any ν to obtain the Bayes error prediction for GP regression with a globally normalised kernel.

Comparisons between the cavity prediction for the learning curves, numerically exact simulated learning curves and the results of the eigenvalue approximation are shown in Figure 3 (left, centre and right), for regular, Erdős-Rényi and generalised random graphs with power law degree distributions respectively. As can be seen the cavity predictions greatly outperform the eigenvalue approximation and are accurate along the whole length of the curve. This confirms our expectation that the cavity approach will become exact on large graphs, although it is remarkable that the agreement is quantitatively so good already for graphs with only five hundred vertices.

4.2 Predicting Prior Variances

As a by-product of the cavity analysis for globally normalised kernels we note that in the cavity form of the Bayes error in Equation (26), the fraction $(\gamma/\sigma^2 + d\kappa (M_d^{-1})_{00})^{-1}$ is the local Bayes error, that is, the local posterior variance. By keeping track of individual samples for this quantity from the population dynamics approach, we can thus predict the distribution of local posterior variances. If we set $\nu = 0$, then this becomes the distribution of prior variances. The cavity approach therefore gives us, without additional effort, a prediction for this distribution.

We can now go back to Section 3.2 and compare the cavity predictions to numerically simulated distributions of prior variances. The cavity predictions for these distributions are shown by the black lines in Figure 2. The cavity approach provides, in particular, detailed information about the tail of the distributions as shown in the insets. There is good agreement between the predictions and the numerical simulations, both regarding the general shape of the variance distributions and the fine structure with a number of non-trivial peaks and troughs. The residual small shifts between the predictions and the numerical results for a single instance of a large graph are most likely due to finite size effects: in a finite graph, the assumption of a tree-like local structure is not exact because there can be rare short cycles; also, the long cycles that the cavity method ignores because their length diverges logarithmically with V will have an effect when V is finite.

4.3 Local Normalisation

We now extend the cavity analysis for the learning curves to the case of locally normalised random walk kernels, which, as argued above, provide more plausible probabilistic models. In this case the diagonal entries of the normalisation matrix $\mathcal{K}$ are defined as

$$\kappa_i = \int \mathrm{d}\boldsymbol{f}\, f_i^2 P(\boldsymbol{f}),$$

where $P(\boldsymbol{f})$ is the GP prior with the unnormalised kernel $\hat{C}$. This makes clear why the locally normalised kernel case is more challenging technically: we cannot calculate the normalisation constants once and for all for a given random graph ensemble and set of kernel parameters p and a as we did for κ in the globally normalised scenario. Instead we have to account for the dependence of the κ_i on the specific graph instance.

On a single graph instance, this stumbling block can be overcome as follows. One iterates the cavity updates (22) for the unnormalised kernel and without the training data (i.e., setting $\kappa = 1$ and $\gamma_i = 0$). The local Bayes error at vertex i, given by the i-th term in the sum from (20), then gives us κ_i. Because $\gamma_i = 0$, one has to use the Woodbury trick to get well-behaved expressions in the limit where the auxiliary parameter $\lambda \to 0$, as explained after (25).

Once the κ_i have been determined in this way, one can use them for predicting the Bayes error for the scenario we really want to study, that is, using a locally normalised kernel and incorporating the training data. The relevant partition function is the analogue of (18) for local normalisation. Dropping the prefactors, the resulting Z can be written as

$$
Z \propto \int \prod_{q=0}^{p} \mathrm{d}\boldsymbol{h}^q \prod_{q=1}^{p} \mathrm{d}\hat{\boldsymbol{h}}^q \prod_i \exp\left(-\frac{1}{2}\sum_{q=0}^{p} c_q d_i h_i^0 h_i^q - \frac{1}{2}\frac{d_i \kappa_i (h_i^0)^2}{\gamma_i/\sigma^2 + \lambda} + \mathrm{i}\sum_{q=1}^{p} d_i \hat{h}_i^q h_i^q \right)
$$
$$
\prod_{(i,j)} \exp\left(-\mathrm{i}\sum_{q=1}^{p} (\hat{h}_i^q h_j^{q-1} + \hat{h}_j^q h_i^{q-1}) \right),
$$

where we have rescaled h_i^q to $d_i^{1/2}\kappa_i^{1/2}h_i^q$ and $\hat{h}_i^q$ to $d_i^{1/2}\kappa_i^{-1/2}\hat{h}_i^q$. Given that the κ_i have already been determined, this is a graphical model for which marginals can be calculated by iterating to a fixed point the equations for the cavity marginals:

$$
P_{\mathrm{loc},j}^{(i)}(\boldsymbol{h}_j, \hat{\boldsymbol{h}}_j | \boldsymbol{x}) \propto \exp\left(-\frac{1}{2}\sum_{q=0}^{p} c_q d_j h_j^0 h_j^q - \frac{1}{2}\frac{d_j \kappa_j (h_j^0)^2}{\gamma_j/\sigma^2 + \lambda} + \mathrm{i}\sum_{q=1}^{p} d_j \hat{h}_j^q h_j^q \right)
$$
$$
\int \prod_{k \in \mathcal{N}(j)\setminus i} \mathrm{d}\boldsymbol{h}_k \mathrm{d}\hat{\boldsymbol{h}}_k \exp\left(-\mathrm{i}\sum_{q=1}^{p} (\hat{h}_j^q h_k^{q-1} + \hat{h}_k^q h_j^{q-1}) \right) P_{\mathrm{loc},k}^{(j)}(\boldsymbol{h}_k, \hat{\boldsymbol{h}}_k | \boldsymbol{x}). \quad (27)
$$

As in Section 4.1 these update equations are solved by cavity marginals of complex Gaussian form, and so we can simplify them to updates for the covariance matrices:

$$
\boldsymbol{V}_{\mathrm{loc},j}^{(i)} = \left(\boldsymbol{O}_{\mathrm{loc},j} - \sum_{k \in \mathcal{N}(j)\setminus i} \boldsymbol{X}\boldsymbol{V}_{k,\mathrm{loc}}^{(j)}\boldsymbol{X} \right)^{-1}. \quad (28)
$$

Here $\boldsymbol{X}$ is defined as in Equation (23) and $\boldsymbol{O}_{\mathrm{loc},j}$ is the obvious analogue of $\boldsymbol{O}_j$ also defined in Equation (23); specifically, κ is replaced by κ_j. Once the update equations have converged, one can calculate the Bayes error from a similarly adapted version of (20).

The above procedure for a single fixed graph now has to be extended to the case of an ensemble of large random graphs characterised by some degree distribution $p(d)$. The outcome of the first round of cavity updates, for the unnormalised kernel without training data, is then represented by a distribution of cavity covariances $\boldsymbol{V}$, while the second one gives a distribution of cavity covariances $\boldsymbol{V}_{\mathrm{loc}}$ for the locally normalised kernel, with training data included. Importantly, these message distributions are coupled to each other via the graph structure, so we need to look at the joint distribution $W(\boldsymbol{V}_{\mathrm{loc}}, \boldsymbol{V})$.

Detailed analysis using the replica method (Urry and Sollich, 2012) shows that the correct fixed point equation updates the $\boldsymbol{V}$-messages as in the globally normalised case with $\gamma = 0$. The second set of local covariances, $\boldsymbol{V}_{\mathrm{loc}}$, are then updated according to (28), with a normaliser calculated using the marginals from the $d-1$ $\boldsymbol{V}$-covariances and an additional 'counterflow' covariance generated from $W(\boldsymbol{V}) = \int d\boldsymbol{V}_{\mathrm{loc}}W(\boldsymbol{V}_{\mathrm{loc}}, \boldsymbol{V})$, subject to the constraint that the local marginals of the neighbours are consistent. We find in practice that the consistency constraint can be dropped and the fixed point

equation for the distribution of the two sets of messages can be approximated by

$$
W(\boldsymbol{V}_{\text{loc}}, \boldsymbol{V}) = \left\langle \sum_d \frac{p(d)d}{\bar{d}} \int \prod_{k=1}^{d-1} \mathrm{d}\boldsymbol{V}_k \mathrm{d}\boldsymbol{V}_{\text{loc},k} \mathrm{d}\boldsymbol{V}_d \prod_{k=1}^{d-1} W(\boldsymbol{V}_{\text{loc},k}, \boldsymbol{V}_k) W(\boldsymbol{V}_d) \right.
$$
$$
\left. \delta\left(\boldsymbol{V}_{\text{loc}} - \left(\boldsymbol{O}_{\text{loc},j} - \sum_{k=1}^{d-1} \boldsymbol{X} \boldsymbol{V}_{k,\text{loc}}^{(j)} \boldsymbol{X} \right)^{-1} \right) \delta\left(\boldsymbol{V} - \left(\boldsymbol{O} - \sum_{k=1}^{d-1} \boldsymbol{X} \boldsymbol{V}_k \boldsymbol{X} \right)^{-1} \right) \right\rangle_{\gamma}. \quad (29)
$$

One sees that if one marginalises over $\boldsymbol{V}_{\text{loc}}$, then one obtains exactly the same condition on $W(\boldsymbol{V})$ as before in the globally normalised kernel case (but with $\kappa = 1$ and $\nu = 0$), see (24). This reflects the fact that the cavity updates for the first set of messages on a single graph do not rely on any information about the second set. The first delta function in (29) corresponds to the fixed point condition for this second set of cavity updates. This condition depends, via the value of the local κ, on the $\boldsymbol{V}$-cavity covariances:

$$
\kappa = \frac{1}{d(\boldsymbol{M}_d^{-1})_{00}}. \quad (30)
$$

It may seem unusual that d copies of $\boldsymbol{V}$ enter here; $\boldsymbol{V}_d$ represents the cavity covariance from the first set that is *received* from the vertex to which the new message $\boldsymbol{V}_{\text{loc}}$ is being *sent*. While this counterflow appears to run against the basic construction of the cavity or belief propagation method, it makes sense here because the first set of cavity messages (or equivalently the distribution $W(\boldsymbol{V})$) reaches a fixed point that is independent of the second set, so the counterflow of information is only apparent. The reason why knowledge about $\boldsymbol{V}_d$ is needed in the update is that κ is the variance of a full marginal rather than a cavity marginal.

Similarly to the case of global normalisation, (29) can be solved by looking for a fixed point of $W(\boldsymbol{V}_{\text{loc}}, \boldsymbol{V})$ using population dynamics. Updates are made by first updating $\boldsymbol{V}$ using Equation (21) and then updating $\boldsymbol{V}_{\text{loc}}$ using (27) with $\kappa \equiv \kappa_i$ replaced by (30).

Once a fixed point has been calculated for the covariance distribution we apply the Woodbury formula to (20) in a similar manner to Section 4.1 to give the prediction for the learning curve for GP regression with a locally normalised kernel. The result for the Bayes error becomes

$$
\varepsilon = \left\langle \sum_d p(d) \int \prod_{k=1}^{d} \mathrm{d}\boldsymbol{V}_{\text{loc},k} \mathrm{d}\boldsymbol{V}_k \prod_{k=1}^{d} W(\boldsymbol{V}_{\text{loc},k}, \boldsymbol{V}_k) \frac{1}{\gamma/\sigma^2 + (\boldsymbol{M}_{d,\text{loc}}^{-1})_{00}/(\boldsymbol{M}_d^{-1})_{00}} \right\rangle_{\gamma}.
$$

Learning curve predictions for GPs with locally normalised kernels as they result from the cavity approach described above are shown in Figure 5. The figure shows numerically simulated learning curves and the cavity prediction, both for Erdős-Rényi random graphs (left) and power law generalised random graphs (centre) of size $V = 500$. As for the globally normalised case one sees that the cavity predictions are quantitatively very accurate even with the simplified update Equation (29). They capture all aspects of learning curve both qualitatively and quantitatively, including, for example, the shoulder in the curves from disconnected single vertices, a feature discussed in more detail below.

The fact that the cavity predictions of the learning curve for a locally normalised kernel are indistinguishable from the numerically simulated learning curves in Figure 5 leads us to believe that the simplification made by dropping the consistency requirement in (29) is in fact exact. This is further substantiated by looking not just at the average of the posterior variance over vertices, which
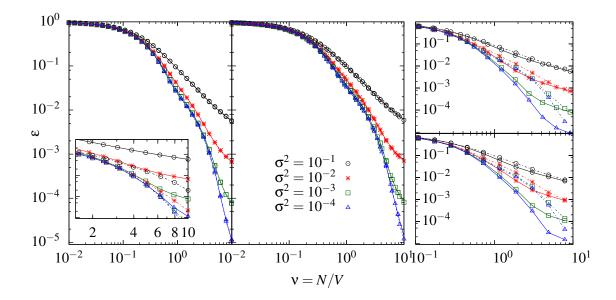
Figure 5: (Left) Learning curves for GP regression with locally normalised kernels with $p = 10$, $a = 2$ on Erdős-Rényi random graphs with mean degree 3, for a range of noise levels σ^2. Solid lines: numerically simulated learning curves for graphs of size $V = 500$, dashed lines: cavity predictions (see Section 4.1); note these are mostly visually indistinguishable from the simulation results. (Left inset) Dotted lines show single vertex contributions to the learning curve (solid line). (Centre) As (left) for power law generalised random graphs with exponent 2.5 and cut off 2. (Right top) Comparison between learning curves for locally (dashed line) and globally (solid line) normalised kernels for Erdős-Rényi random graphs. (Right bottom) As (right top) for power law random graphs.

is the Bayes error, but its distribution across vertices. As shown in Figure 6, the cavity predictions for this distribution are in very good agreement with the results of numerical simulations. This holds not only for the two values of ν shown, but along the entire learning curve.

5. A Qualitative Comparison of Learning with Locally and Globally Normalised Kernels

The cavity approach we have developed gives very accurate predictions for learning curves for GP regression on graphs using random walk kernels. This is true for both global and local normalisations of the kernel. We argued in Section 3.2 that the local normalisation is much more plausible as a probabilistic model, because it avoids variability in the local prior variances that is non-trivially related to the local graph structure and so difficult to justify from prior knowledge. We now compare what the qualitative effects of the two different normalisations are on GP learning.

It is not a simple matter to say which kernel is 'better', the locally or globally normalised one. Since we have dealt with the matched case, where for each kernel the target functions are sampled from a GP prior with that kernel as covariance function, it would not make sense to say the better kernel is the one that gives the lower Bayes error for given number of examples, as the Bayes error
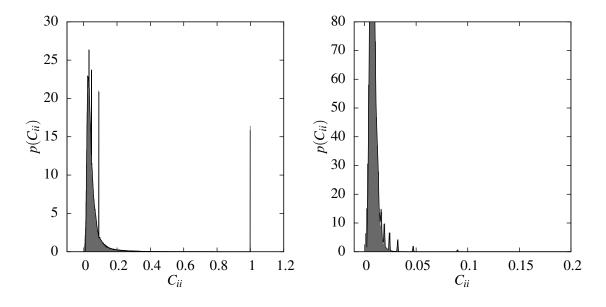
Figure 6: (Left) Grey: histogram of posterior variances at $v = 1.172$ for the locally normalised random walk kernel with $a = 2$, $p = 10$, averaged over ten samples each of teacher functions, data and Erdős-Rényi graphs with mean degree $\lambda = 3$ and $V = 1000$ vertices. Black: cavity prediction for this distribution in the large graph limit. (Right) As (left) but for $v = 6.210$.

reflects both the complexity of the target function and the success in learning it. A more definite answer could be obtained only empirically, by running GP regression with local and global kernel normalisation on the same data sets and comparing the prediction errors and also the marginal data likelihood. The same approach could also be tried with synthetic data sets generated from GP priors that are mismatched to both priors we have considered, defined by the globally and locally normalised kernel, though justifying what is a reasonable choice for the prior of the target function would not be easy.

While a detailed study along the lines above is outside the scope of this paper, we can nevertheless at least qualitatively study the effect of the kernel normalisation, to understand to what extent the corresponding priors define significantly different probabilistic models. Figure 5 (right top and bottom) overlays the learning curves for global and local kernel normalisations, for an Erdős-Rényi and a power law generalised random graph respectively. There are qualitative differences in the shapes of the learning curves, with the ones for the locally normalised kernel exhibiting a shoulder around $v = 2$. This shoulder is due to the proper normalisation of isolated vertices to unit prior variance; by contrast, as shown earlier in Figure 2 (left), global normalisation gives too small a prior variance to such vertices. The inset in Figure 5 (left) shows the expected learning curve contributions from all locally normalised isolated vertices (single vertex subgraphs) as dotted lines. After the GP learns the rest of the graph to a sufficient accuracy, the single vertex error dominates the learning curve until these vertices have typically seen at least one example. Once this point has been passed, the dominant error comes once more from the giant connected component of the graph, and the GP

learns in a similar manner to the globally normalised case. A similar effect, although not plotted, is seen for the generalised random graph case.

We can extend the scope of this qualitative comparison by examining how a student GP with a kernel with one normalisation performs when learning from a teacher with a kernel with the other normalisation. This is a case of *model mismatch*; our theory so far does not extend to this scenario, but we can obtain learning curves by numerical simulation. Figure 7 (left) shows the case of GP students with a globally normalised kernel learning from a teacher with a locally normalised kernel on an Erdős-Rényi graph. The learning curves for the mismatched scenario are very different from those for the matched case (Figures 3 and 5), showing an increase in error as ν approaches unity. The resulting maximum in the learning curve again emphasises that the two choices of normalisation produce distinctly different probabilistic models. Similar behaviour can be observed for the case of power law generalised random graphs, shown in Figure 7 (right) and for the case of GP students with a locally normalised kernel learning from a teacher with a globally normalised kernel, shown in Figure 8. In all cases close inspection (see Appendix B) shows that the error maximum is caused by 'dangling edges' of the graph, that is, chains of vertices (with degree two) extending away from the giant graph component and terminating in a vertex of degree one.

As a final qualitative comparison between globally and locally normalised kernels, Figure 9 shows the variance of local posterior variances. This measures how much the local Bayes error typically varies from vertex-to-vertex, as a function of the data set size. Plausibly one would expect that this error variance is low initially when prediction on all vertices is equally uncertain. For large data sets the same should be true because errors on all vertices are then low. In an intermediate regime the error variance should become larger because examples will have been received on or near some vertices but not others. As Figure 9 shows, for kernels with local normalisation we find exactly this scenario, both for Erdős-Rényi and power law random graphs. The error variance is low for small $\nu = N/V$, increasing to a peak at $\nu \approx 0.2$ and finally decreasing again.

These results can now be contrasted with those for globally normalised kernels, also displayed in Figure 9. Here the error variance is largest at $\nu = 0$ and decays from there. This means that the initial variance in the local prior variances is so large that any effects from the uneven distribution of example locations in any given data set remain sub-dominant throughout. We regard this as another indication of the probabilistically implausible character of the large spread of prior variances caused by global normalisation.

6. Conclusions and Further Work

In this paper we studied random walk kernels and their application to GP regression. We began, in Section 2, by studying the random walk kernel, with a focus on applying this to d-regular trees and graphs. We showed that the kernel exhibits a rather subtle approach to the fully correlated limit; this limit is reached only beyond a graph-size dependent threshold for the kernel range p/a, where cycles become important. If p/a is large but below this threshold, the kernel reaches a non-trivial limiting shape.

In Section 3 we moved on to the application of random walk kernels to GP regression. We showed, in Section 3.2, that the more typical approach to normalisation, that is, scaling the kernel globally to a desired average prior variance, results in a large spread of local prior variances that is related in a complicated manner to the graph structure; this is undesirable in a prior. We suggested
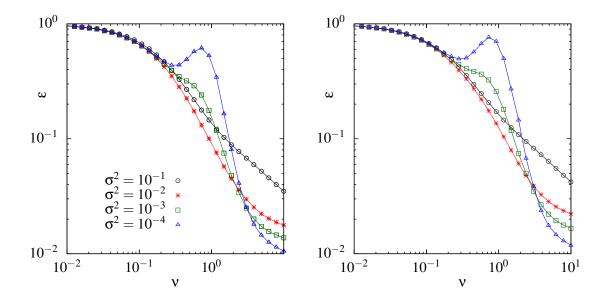
Figure 7: (Left) Numerically simulated learning curves for a GP with a globally normalised kernel with $p = 10$ and $a = 2$, on Erdős-Rényi random graphs with mean degree 3 for a range of noise levels. The teacher GP has a locally normalised kernel with the same parameters. (Right) As (left) but for power law generalised random graphs with exponent 2.5 and cutoff 2.

as a simple remedy to perform local normalisation, where the raw kernel is normalised by its local prior variance so that the prior variance becomes the same at every vertex.

In order to get a deeper understanding of the performance of GPs with random walk kernels we then studied the learning curves, that is, the mean Bayes error as a function of the number of examples. We began in section 3.4 by applying a previous approximation due to Sollich (1999a) and Malzahn and Opper (2005) to the case of discrete inputs that are vertices on a graph. We demonstrated numerically that this approximation is accurate only for small and large number of training examples per vertex, $\nu = N/V$, while it fails in the crossover between these two regimes. The outline derivation of this approximation suggested how one might improve it: one has to exploit fully the structure of random graphs, using the cavity method, thus avoiding approximating the average over data sets. In Section 4 we implemented this programme, beginning in Section 4.1 with the case of global normalisation. We showed that by Fourier transforming the prior and introducing $2p$ additional variables at each vertex one can rewrite the partition function in terms of a complex-valued Gaussian graphical model, where the marginals that are required to calculate the Bayes error can be found using the cavity method, or equivalently belief propagation. In Section 4.3 we tackled the more difficult scenario of a locally normalised kernel. This required two sets of cavity equations. The first serves to calculate the local normalisation factors. The second one then combines these with the information about the data set to find the local marginal distributions. One might be tempted to consider applying our methods to a lattice so that one could make an estimate of the learning curves for the continuous limit, that is, regression with a squared exponential kernel
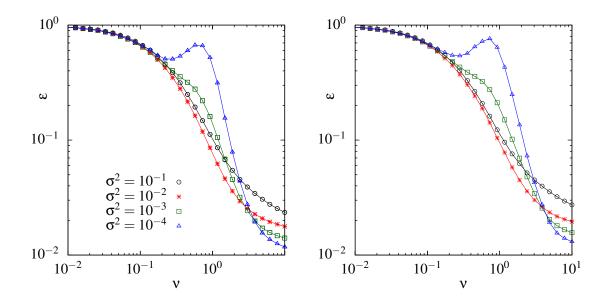
Figure 8: (Left) Numerically simulated learning curves for a GP with a locally normalised kernel with $p = 10$ and $a = 2$, on Erdős-Rényi random graphs with mean degree 3 for a range of noise levels. The teacher GP has a globally normalised kernel with the same parameters. (Right) As (left) but for power law generalised random graphs with exponent 2.5 and cutoff 2.
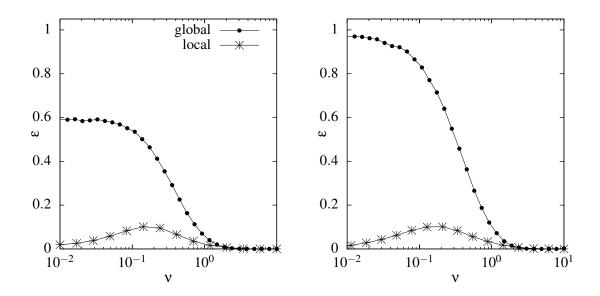

Figure 9: (Left) Error variance for GP regression with locally (stars) and globally (circles) normalised kernels against ν, on Erdős-Rényi random graphs, and for matched learning with $p = 10$, $a = 2$, $\sigma^2 = 0.1$. (Right) As (left) but for power law generalised random graphs with exponent 2.5 and cutoff 2.

for inputs in $\mathbb{R}^2$ or similar. Sadly, however, since the cavity method requires graphs to be treelike this is not possible.

Finally in Section 5 we qualitatively compared GPs with kernels that are normalised globally and locally. We showed that learning curves are indeed qualitatively different. In particular, local normalisation leads to a shoulder in the learning curves owing to the correct normalisation of the single vertex disconnected graph components. We also considered numerically calculated mismatch curves. The mismatch caused a maximum to appear in the learning curves, as a result of the large differences in the teacher and student priors. Lastly we looked at the variance among the local Bayes errors, for GPs with both globally and locally normalised kernels. Plausibly, locally normalised kernels lead to this error variance being maximal for intermediate data set sizes. This reflects the variation in number of examples seen at or near individual vertices. For globally normalised kernels, the error variance inherited from the spread of local prior variances is always dominant, obscuring any signatures from the changing 'coverage' of the graph by examples.

In further work we intend to extend the cavity approximation of the learning curves to the case of mismatch, where teacher and student have different kernel hyperparameters. It would also be interesting to apply the cavity method to the learning curves of GPs with random walk kernels on more general random graphs, like those considered in Rogers et al. (2010) and Kühn and van Mourik (2011). This would enable us to consider graphs exhibiting some community structure. Looking further ahead, preliminary work has shown that it should be possible to extend the cavity learning curve approximation to the problem of graph mismatch, where the student has incomplete information about the graph structure of the teacher.

Appendix A. Random Walk Kernel on d-Regular Trees

We detail here how to derive the limiting kernel from (5) in Section 2.1 and how to calculate the scaling of the learning curve stated in Section 3.4.1.

A.1 Large-p Limit from Heat Kernel Results

In Chung and Yau (1999) the authors considered heat kernels on graphs. The heat kernel for a graph with normalised Laplacian L (see Section 2 in the main text) is given by

$$H = \exp(-tL), \quad t > 0,$$

This is exactly the diffusion kernel given in (1) with $t = \frac{1}{2}\sigma^2$. Chung and Yau (1999) gave results for heat kernels on d-regular trees. Their method involves treating the tree as a covering of a path, which is closely related to the mapping onto a one-dimensional lattice discussed in Section 2.1. The eigenvectors and eigenvalues of L are then found within this mapping. If we call the unnormalised form of the random walk kernels we considered in the main text again $\hat{C} = \left(I - a^{-1}L\right)^p$, then we can directly use the results of Chung and Yau (1999), simply by modifying the function that is applied to each eigenvalue λ of L from $\exp(-t\lambda)$ to $(1 - \lambda/a)^p$. This gives, for the unnormalised

random walk kernel element between two vertices on the tree at distance l,

$$\hat{C}_{0,p} = \frac{2(d-1)d}{\pi} \int_0^{\pi} dx \, \frac{[1 - (1 - 2\cos(x)\sqrt{d-1}/d)/a]^p \sin^2(x)}{d^2 - 4(d-1)\cos^2(x)},$$

$$\hat{C}_{l \geq 1,p} = \frac{2}{\pi(d-1)^{l/2-1}} \int_0^{\pi} dx \, [1 - (1 - 2\cos(x)\sqrt{d-1}/d)/a]^p$$

$$\times \frac{\sin(x)[(d-1)\sin((l+1)x) - \sin((l-1)x)]}{d^2 - 4(d-1)\cos^2(x)}. \tag{31}$$

In the limit $p \to \infty$, the factor in square brackets in both expressions means that only values of x up to $O(p^{-1/2})$ contribute significantly to the integral. One can then expand $\sin(x) \approx x$ linearly everywhere, and similarly use $\cos(x) \approx 1$. Calculating $\hat{C}_{l,p}$ in this way for $p \to \infty$ gives the result (5) for $C_{l,p} = \hat{C}_{l,p}/\hat{C}_{0,p}$.

A.2 Scaling Form for Large p

In Section 3.4.1 we derived the learning curve decay due to the first example, given in (15). As explained there, to understand how this behaves for large p one needs to know how $C_{l,p}$ approaches its limiting form (5). In the outline derivation in the previous subsection we saw that in order to obtain this limiting form one expands not just $\sin(x)$, but also $\sin((l-1)x)$ and $\sin((l+1)x)$ linearly because $x = O(p^{-1/2})$ is small. For the two sine factors this will break down once $lp^{-1/2}$ is of order one, that is, for large enough l. This suggests looking at $C_{l,p}$ as function of $l' = lp^{-1/2}$. To focus on the relevant part of the integral one can similarly transform the integration variable to $x' = xp^{1/2}$. For $p \to \infty$ at constant l', the integral for $\hat{C}_{l,p}$ in (31) then becomes proportional to

$$(d-1)^{-l/2} \int_0^{\infty} dx' \, e^{-x'^2 \sqrt{d-1}/(da)} x'(d-2) \sin(l'x').$$

Integration by parts now gives $C_{l,p} \propto \hat{C}_{l,p} \propto (d-1)^{-l/2} l' \exp(-\frac{(l')^2 d}{2\sqrt{d-1}})$. Comparing with the large-l limit of (5), $C_{l,p} \propto (d-1)^{-l/2} l'$, shows that the effect of finite p is contained in the exponential (Gaussian) cutoff factor which becomes significant for l' of order unity, that is, $l = O(p^{1/2})$, as stated in the main text.

Visually, the cutoff effects for large l and p can be seen more clearly by plotting not $C_{l,p}$ directly but rather $\hat{R}_{l,p}$, the unormalised version of $R_{l,p}$ as defined in Section 3.4.1. As $v_l = d(d-1)^{l-1}$ for $l \geq 1$, the additional $\sqrt{v_l}$ factor just removes the decay with $(d-1)^{-l/2}$ from $C_{l,p}$. From the above discussion, we then expect to find for large l that $\hat{R}_{l,p} \propto l' \exp(-\frac{(l')^2 d}{2\sqrt{d-1}})$. A plot of numerical results for a suitably normalised version of $\hat{R}_{l,p}$ (see below), plotted against $lp^{-1/2}$ for increasing p, is shown in Figure 10. There is a clear trend towards the predicted asymptotic behaviour for large p (dashed line).

It may be somewhat surprising that the cutoff lengthscale that appears in the analysis above is $l = O(p^{1/2})$, which for large p is much smaller than the typical kernel range p/a. To understand this intuitively, one can go back to (6) for $R_{l,p}$ and use that this is essentially unnormalised diffusion. Taking $\hat{R}_{l,p}$ as the unnormalised version of $R_{l,p}$ again and letting $\rho_{l,p} = \hat{R}_{l,p}\gamma^{-p}$ with $\gamma = (1 - 1/a) + 2\sqrt{d-1}/(ad)$ to re-establish normalisation, one finds that $\rho_{l,p}$ evolves almost according to a diffusion process in 'time' p, except that probability conservation is broken at $l = 0$ and $l = 1$. In the (leaky) diffusion process $\rho_{l,p}$ the typical lengthscale l should then scale with time as $p^{1/2}$, exactly
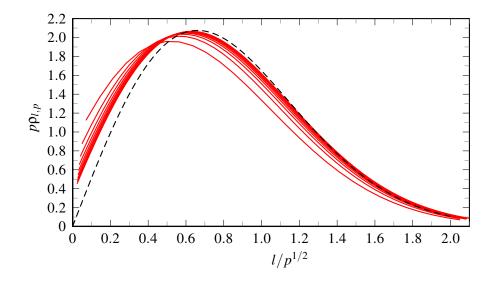
Figure 10: Plot demonstrating the $p \to \infty$ scaling from of $p\rho_{l,p}$, the normalised version of $\hat{R}_{l,p}$. Dashed line shows the analytically found functional from of the scaling.

as we found above. This diffusion interpretation can also be used to reproduce the quantitative scaling form, by adapting the methods of Monthus and Texier (1996) where the authors study a one dimensional walk with a reflective boundary to compute the large-p scaling. The normalised version of $\hat{R}_{l,p}$ shown in Figure 10 is in fact $p\rho_{l,p}$.

Appendix B. Analysis of the Mismatch Learning Curves

In this appendix we suggest a way of understanding the mismatch maximum seen in the learning curves plotted in Figure 7 (left) and (right) and Figure 8 (left) and (right). We begin by considering the mean of the GP posterior, that is, the prediction function. If we arrange the predictive means at each vertex into a vector $\bar{f}$, then from (8) this can be written as

$$\bar{f} = K_V K^{-1} y,$$

where $(K_V)_{jv} = C_{j,x_v}$ for $v = 1, \ldots, N$ and $j = 1, \ldots, V$ and $K_{\mu v} = C_{x_\mu, x_v} + \sigma^2 \delta_{\mu, v}$ for $\mu, v = 1, \ldots, N$. (Recall that x_μ is the location of the μ-th training input, which on a graph is just a label in the range $\{1, \ldots, V\}$.) We can rewrite this in the form $\bar{f} = Mz$ where

$$M = K_V K^{-1/2},$$
$$z = K^{-1/2} y.$$

One sees that M represents teacher-independent aspects of $\bar{f}$. The vector of 'pseudo-training outputs' z has been defined so that in the *matched* case, it obeys $\langle zz^T \rangle = I$. We think of z as pseudo-training outputs because if its components z_μ are sampled as independent and identically distributed unit variance Gaussian random variables, then $\bar{f} = Mz$ will have the correct distribution of mean prediction vectors. The columns $m_1, \ldots, m_N$ of $M = \begin{pmatrix} m_1 & \cdots & m_N \end{pmatrix}$ represent the 'effective
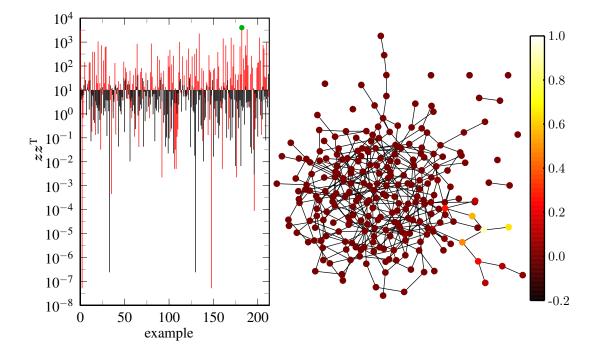
Figure 11: (Left) Black lines show diagonal elements of zz^{T} for a typical draw from the Erdős-Rényi random graph ensemble with average degree 3 in a matched scenario with 250 vertices, $p = 10$, $a = 2$ and $\sigma^2 = 0.0001$. Red lines show zz^{T} for the case where student is globally normalised and the teacher is locally normalised. (Right) A plot of the 'prediction vector' for the largest spike in zz^{T} (indicated by a green circle in the left hand plot) under mismatch.

prediction vectors' conjugate to the z_μ. In the matched case each $\bar{f}_i^2$ will then be, on average, a sum of squares of the corresponding entries in the effective prediction vectors.

For the case of mismatch, for example, because student and teacher use differently normalised kernels, the only change is in the statistics of z. Their covariance matrix $\langle zz^{\mathrm{T}} \rangle$ will no longer simply be the identity matrix, and so act as a re-weighting of the prediction vectors. To understand our numerical simulations, we considered a value of ν around the maximum in the mismatched learning curve, listed the largest (diagonal) entries of zz^{T} and plotted their corresponding prediction vectors. These prediction vectors were generally localised around dangling ends of the graph, substantiating our claim in the main text that it is from these graph regions that the major contribution to the learning curve maximum arises. Typical plots of zz^T and the prediction vector corresponding to the largest spike in zz^{T} for an instance of an Erdős-Rényi graph with 250 vertices are shown in Figures 11 and 12 for both a locally normalised teacher and globally normalised student and a globally normalised teacher and locally normalised student respectively. Both these plots clearly show the largest spike corresponding to a prediction vector localised on a dangling edge in the graph. Similar plots of the other large spikes give prediction vectors localised on other dangling edges.
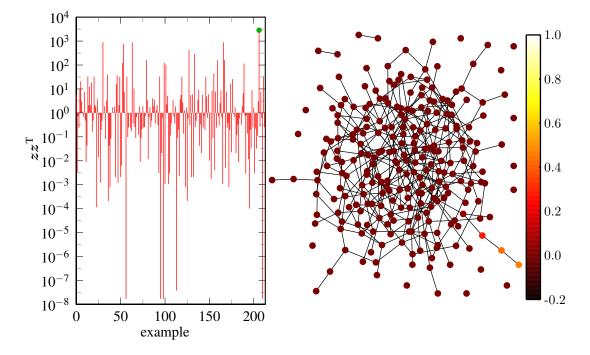
Figure 12: Analogue of Figure 11 for a locally normalised student and globally normalised teacher

References

S. Amari, N. Fujita, and S. Shinomoto. Four types of learning curves. *Neural Comput.*, 4(4): 605–618, 1992.

C. Bishop. *Pattern Recognition And Machine Learning*. Springer, New York, NY, USA., 2nd edition, 2007.

T. Britton, M. Deijfen, and A. Martin-Loeff. Generating simple random graphs with prescribed degree distribution. *J. Stat. Phys.*, 124(6):1377–1397, 2006.

F. K. Chung. *Spectral Graph Theory*, volume 92 of *Regional Conference Series In Mathematics*. American Mathematical Society, Dec 1996.

F. K. Chung and S. T. Yau. Coverings, heat kernels and spanning trees. *Electronic Journal Of Combinatorics*, 6:R12, 1999.

N. Cristianini and J. Shawe-Taylor. *An Introduction To Support Vector Machines And Other Kernel-Based Learning Methods*. Cambridge University Press, 2000.

P. Erdős and A. Rényi. On random graphs, I. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

J. Freeman and D. Saad. Dynamics of on-line learning in radial basis networks. *Phys. Rev. E*, 56 (1):907–918, Jul 1997.

M. G. Genton. Classes of kernels for machine learning: A statistics perspective. *Journal Of Machine Learning Research*, 2(2):299–312, 2002.

W.W. Hager. Updating the inverse of a matrix. *SIAM Rev.*, 31(2):221–239, 1989.

M. S. Handcock and M. L. Stein. A Bayesian analysis of Kriging. *Technometrics*, 35(4):403–410, 1993.

D. Haussler, M. Kearns, H. S. Seung, and N. Tishby. Rigorous learning curve bounds from statistical mechanics. *Mach. Learn.*, 25(2–3):195–236, 1996.

J. P. C. Kleijnen. Kriging metamodeling in simulation: A review. *European Journal Of Operational Research*, 192(3):707 – 716, 2009.

R. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. In C. Sammut and Hoffmann A. G., editors, *Proceedings Of The 19th International Conference On Machine Learning (ICML)*, pages 315–322, San Francisco, CA, USA., 2002. Morgan Kaufmann.

R. Kühn and J. van Mourik. Spectra of modular and small-world matrices. *J. Phys. A Math. Gen.*, 44(16):165205, 2011.

D. Malzahn and M. Opper. Learning curves and bootstrap estimates for inference with Gaussian processes: A statistical mechanics study. *Complexity*, 8(4):57–63, 2003.

D. Malzahn and M. Opper. A statistical physics approach for the analysis of machine learning algorithms on real data. *J. Stat. Mech. Theor. Exp.*, 2005(11):P11001, 2005.

B. D. McKay. The expected eigenvalue distribution of a large regular graph. *Linear Algebra And Its Applications*, 40:203–216, 1981.

J. R. Meinhold and N. D. Singpurwalla. Understanding the Kalman filter. *The American Statistician*, 37(2):123–127, 1983.

M. Mézard and G. Parisi. The Bethe lattice spin glass revisited. *Eur. Phys. J. B*, 20(2):217–233, 2001.

M. Mézard and G. Parisi. The cavity method at zero temperature. *J. Stat. Phys.*, 111(1–2):1–34, 2003.

M. Mézard, G. Parisi, and M. Virasoro. *Spin Glass Theory And Beyond*, volume 9 of *World Scientific Lecture Notes In Physics*. World Scientific, Singapore, 1987.

C. Monthus and C. Texier. Random walk on the Bethe lattice and hyperbolic Brownian motion. *J. Phys. A Math. Gen.*, 29:2399–2409, 1996.

K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions On Neural Networks*, 12(2):181–201, 2001.

R. M. Neal. *Bayesian Learning For Neural Networks*, volume 118 of *Lecture Notes In Statistics*. Springer-Verlag, Berlin, Germany, 1996.

M. Opper and D. Haussler. Bounds for predictive errors in the statistical mechanics of supervised learning. *Phys. Rev. Lett.*, 75(20):3772–3775, 1995.

M. Opper and D. Malzahn. A variational approach to learning curves. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances In Neural Information Processing Systems*, volume 14, pages 463–469, Cambridge, MA, USA., 2002. The MIT Press.

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

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes For Machine Learning*. MIT Press, Nov 2005.

T. Rogers, C. Vicente, K. Takeda, and I. Castillo. Spectral density of random graphs with topological constraints. *J. Phys. A Math. Gen.*, 43(19):195002, 2010.

H. S. Seung, H. Sompolinsky, and N. Tishby. Statistical mechanics of learning from examples. *Phys. Rev. A*, 45(8):6056–6091, 1992.

A. Smola and R. Kondor. Kernels and regularization on graphs. In B Schölkopf and M. K. Warmuth, editors, *Lect. Notes Artif. Int.*, volume 2777, pages 144–158, 2003.

P. Sollich. Learning curves for Gaussian processes. In M. S. Kearns, S. A. Solla, and D A Cohn, editors, *Advances In Neural Information Processing Systems*, volume 11, pages 344–350, Cambridge, MA, USA., 1999a. The MIT Press.

P. Sollich. Approximate learning curves for Gaussian processes. In *Ninth International Conference On Artificial Neural Networks (ICANN99)*, pages 437–442, Edison, NJ, USA., 1999b. Institute of Electrical Engineers Inspec inc.

P. Sollich. Gaussian process regression with mismatched models. In S Becker T G Dietterich and Z Ghahramani, editors, *Advances In Neural Information Processing Systems*, volume 14, pages 519–526, Cambridge, MA, USA., 2002. The MIT Press.

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

P. Sollich and C. K. I. Williams. Using the equivalent kernel to understand Gaussian process regression. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances In Neural Information Processing Systems*, volume 17, pages 1313–1320, Cambridge, MA, USA., 2005. The MIT Press.

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*, volume 22, pages 1723–1731, Red Hook, NY, USA., 2009. Curran Associates Inc.

M. J. Urry and P. Sollich. Exact learning curves for Gaussian process regression on large random graphs. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances In Neural Information Processing Systems*, volume 23, pages 2316–2324, Red Hook, NY, USA., 2010. Curran Associates Inc.

M. J. Urry and P. Sollich. Replica theory for learning curves for gaussian processes on random graphs. *J. Phys. A Math. Gen.*, 45(42):425005, 2012.

T. L. H. Watkin, A. Rau, and M. Biehl. The statistical mechanics of learning a rule. *Rev. Mod. Phys.*, 65(2):499–556, 1993.

C. Williams and F. Vivarelli. Upper and lower bounds on the learning curve for Gaussian processes. *Mach. Learn.*, 40:77–102, 2000.