杭州電子科技大学

硕 士 学 位 论 文

题　　目：大数据高斯过程回归模型研究

研 究 生　周昌凯

专　　业　应用统计

指导教师　王文胜 教授

企业导师　许亦频

完成日期　2022 年 03 月

杭州电子科技大学硕士学位论文

大数据高斯过程回归模型研究

研 究 生：周昌凯

指导教师：王文胜　教授

2022 年 03 月

Dissertation Submitted to Hangzhou Dianzi University for the Degree of Master

Gaussian Process Regression Research for Big Data

Candidate： **Changkai Zhou**

Supervisor： **Prof. Wensheng Wang**

March，2022

杭州电子科技大学

学位论文原创性声明和使用授权说明

原创性声明

本人郑重声明： 所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

论文作者签名： 日期： 年 月 日

学位论文使用授权说明

本人完全了解杭州电子科技大学关于保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属杭州电子科技大学。本人保证毕业离校后，发表论文或使用论文工作成果时署名单位仍然为杭州电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。（保密论文在解密后遵守此规定）

论文作者签名： 日期： 年 月 日

指导教师签名： 日期： 年 月 日

摘 要

近二三十年，机器学习中最活跃的研究方向之一是开发实用的贝叶斯方法来解决“学习”问题。高斯过程在机器学习领域的应用展现出了一种最重要的贝叶斯机器学习方法，即它是基于给定函数空间上先验分布的有效办法。同样，作为核方法，高斯过程为机器学习提供了一个有依据的、实用的、概率性的框架。长期的理论与应用的发展使高斯过程在解释性方面具有优势，并为学习和模型选择提供了一个有依据的框架，最终使得高斯过程模型在监督学习方面占据重要的地位。

然而完整的高斯过程模型最突出的弱点在于其难以应用于大数据。给定包含 n 个样本的数据集，标准的高斯过程模型在训练过程中的时间复杂度为$\mathcal{O}\left(n^3\right)$，因为需要对$n\times n$的协方差矩阵求逆、求行列式；在预测过程中需要花费$\mathcal{O}\left(n^2\right)$，因为使用矩阵向量乘法去加速该过程。该弱点局限了标准的高斯过程模型难以应用于数据量大小为$\mathcal{O}\left(10^4\right)$的数据集。

高斯过程模型在大数据集上的拓展形式是长久的需求，无论是基于模型本身的限制，还是大数据时代的背景。然而，由于该方向的研究在高斯过程模型被广泛应用以来一直是热门的领域，故本文在总结主流的拓展方法之后，一是基于聚合模型的框架提出双层的在大数据集上可保持一致性的高斯过程回归模型，二是基于分布式异方差稀疏高斯过程模型，研究如何添加诱导点使得近似模型能够保持原模型的精度。其中，一致性理论来源于高斯过程回归模型与克里金插值法的联系，并且高斯过程模型与其他模型的联系(如神经网络)揭露了高斯过程模型一些有趣的性质。

本文的实验基于玩具数据集及大量的现实数据集，在多方面评价改进模型的提升效果。实验结果显示：一为双层聚合模型在大数据上能够保持预测的一致性，在聚合模型类中保持最优的预测精度；二为模型添加诱导点的方法能够还原完整模型的预测能力。

关键词：高斯过程回归，大数据，稀疏近似，多层模型，专家模型

ABSTRACT

In the past two to three decades, one of the most active research directions in machine learning has been the development of practical Bayesian methods to solve "learning" problems. The application of Gaussian process in machine learning shows one of the most important Bayesian machine learning methods, which is an effective method based on the given prior distribution over a function space. Similarly, as a kernel method, the Gaussian process provides a principled, practical, and probabilistic framework for machine learning. The long-term development of theory and application has given the Gaussian process an advantage in terms of interpretability, and has provided a basis for learning and model selection. Finally, the Gaussian process model occupies an important position in supervised learning.

However, the most prominent weakness of the full Gaussian process model is that it is difficult to apply to big data. Given a data set containing n samples, the time complexity of the full Gaussian process model in the training process is $\mathcal{O}\left(n^3\right)$ because the $n\times n$ covariance matrix needs to be inverted and the determinant is calculated; it needs to be spent $\mathcal{O}\left(n^2\right)$ in prediction because the matrix vector multiply is used to speed up the process. This weakness restricts the full Gaussian process model for data sets with $\mathcal{O}\left(10^4\right)$ samples.

The expansion of the Gaussian process model on large data sets is a long-term demand, whether it is based on the limitations of the model itself or the background of the big data era. However, because the research in this direction has been a hot field since the Gaussian process model is widely used, this article summarizes the mainstream expansion methods. First, we based on the aggregation model proposes a two-layer framework that can maintain consistency on large data sets. Gaussian process model. Second, based on the distributed heteroscedasticity sparse Gaussian process model, we study how to add inducing points so that the approximate model can maintain the accuracy of the original model. Among them, the consistency theory comes from the connection between the Gaussian process model and the Kriging method, and the connection between the Gaussian process model and other models (e.g., neural networks) reveals some interesting properties of the Gaussian process model.

The experiment in this paper is based on toy data sets and a large number of real data sets, and evaluates the improvement effect of the improved model in many aspects. The experimental results show that: a two-layer aggregation model can maintain the consistency of prediction on big data, and maintain the best prediction accuracy in the aggregation model category; second, the method of adding induction points to the model can restore the prediction ability of the complete model.

Keywords: Gaussian process regression, big data, sparse approximation, hierarchical model, mixture of experts

目 录

1 绪论

1.1 研究背景与意义

1.1.1 基于自然科学史的研究背景

“回归”（Regression）在统计学上是变量分布向平均值趋近的意思，回归的现象最初由高尔顿（Francis Galton）和皮尔逊（Karl Pearson）于 1886 年《身高遗传向平均回归》一文中正式记录，描述身高低于父辈平均水平的父亲其儿子身高大多超过父亲，而身高高于父辈平均水平的父亲其儿子身高大多矮于父亲，即儿子的身高有向父辈平均身高回归的趋势[1]。其中，儿子的平均身高（约 69 英寸）大于父辈的平均身高（约 68 英寸），说明该部分是由环境等影响导致遗传物质变化的部分，回归现象则说明除前部分外的身高参差不是由遗传造成的，体现为儿子身高与父亲身高的联合分布分别对于儿子身高与父亲身高的两个条件分布是相同的，结合父亲高其儿子普遍高的现象，则可以体现联合分布的协方差不为 0，在二维图中可体现为散点围绕直线（回归方程）呈椭圆形特征。由上可知，回归现象并不依赖于回归方程来体现，而主要和遗传本身相关。换作大麦麦芒长度的例子[2]，得到一片麦穗地上全部麦穗的统计数据，如图 1.1 所示。图中横轴是麦芒的长度，纵轴表示频数。现挑选一组麦芒长度明显超过平均值的麦穗（黑色表示）培育后代，并同样作一张新的统计图。达尔文可能会预测，统计曲线应该整体右移，或者说麦芒的平均长度会增加。但实际情况并没有出现预料中的情况，而和第一次作出的图完全相同，即使选择麦芒特别短的麦穗作种子，情况也没有不同。选种没有产生影响，作图的相同说明这些广泛存在的微小连续变异[1]无法遗传，表现出了回归的现象。

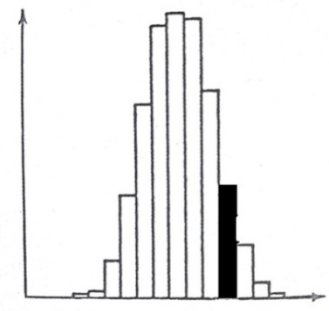

图 1.1 纯种大麦麦芒长度统计图[2]

[1]与微小的连续的不可遗传的变异相对应的是不连续的“跃迁式”的可遗传的突变，比如说突变使得麦穗根本没有麦芒，并且可以通过培育使得后代都呈现无芒的特性。

包含少量原子（1000 个左右）的基因却能表现出近乎奇迹的稳定性，从统计学角度可以得到什么样的启发[2]？使用哈布斯堡王朝（Habsburg Dynasty）的例子，其中的王室成员长着一种特别难看的下嘴唇，令人惊奇的是，通过 16 世纪至 19 世纪的王室后裔肖像，可以发现这种决定异常特征的基因物质结构每次都非常精确地被复制了。这种惊人的精确性，从统计角度反推，可以猜想基因的结构具有高度的稳定性，只有可能是分子。进一步，生命有机体的生命过程也充满着统计的思想。用熵作为系统有序或无序的度量，生命体在大体上遵循着以从无序到有序的统计学趋势为基础的普通物理定律，其中，熵越大表示系统越无序。然而，有机体在生命周期内长期呈现出的有序性与规律性，暗示着生命体通过“负熵”避免了快速衰败至惰性的平衡态。更有趣的是，有序性从何而来？薛定谔（Erwin Schrödinger）将普朗克（Max Planck）的动力学和统计学的规律分别称为“从有序到有序”和“从无序到有序”。前者的模型为诸如行星或时钟等宏观机械运动，而后者则体现为微观方面无序的、难以预测的微粒相互作用，在宏观方面呈现出有序的现象。简单地，用扩散现象作为例子，将紫红色的高锰酸钾加入水中，其独立的分子进行布朗运动的同时也与其他分子碰撞，而从视觉方面则可以观察到溶质从高浓度向低浓度的有序的扩散过程，直至均匀分布在水中。在这个微观充满动力学思想的系统中引入概率，并不是将概率作为一种近似的方法，而是当作一种解释的原则，用概率来证明，假设一个系统是由大量粒子组成的，因而概率是适用的，那么系统可能会显示出一种“新型”的行为。

以动力学为基础的机械论倾向于强调稳定、有序、均匀和平衡，它最关心的是封闭系统与线性关系，输入的小幅度的改变也总是对结果产生小幅度的影响。普里戈金（Ilya Prigogine）则将目光转向了另一个方面：不稳定、无序、非均衡、非线性关系及对时间的高度敏感性[3]。他认为，有序和组织可以通过一个“自组织”的过程从无序和热混沌中“自发的产生出来”。在这个过程中，必然性与偶然性并非是不可协调而相互对立之物，而是在“命运”中各司其职的伙伴关系。简而言之，在稳定的系统中，必然的决定论过程起着主导的作用，犹如子辈身高向父辈身高均值回归的现象，而系统到了不稳定的“分岔”路口时，通常意义下的著名的大数定律被打破了，一个小小的噪声或者说是涨落都可能引起宏观系统的剧烈变化。特别地，对于时间的高度敏感性—不可逆性[2]—可能是有序的源泉，其在统计学上的描述则可以看作是熵增的定律，或者说是不确定性随着时间的流逝而增加的现象。在这其中，对于稳定系统而言，波尔兹曼（Ludwig Boltzmann）解释道，概率可以恰当地解释系统对一切初始复杂分布的遗忘，由系统到无序的演化，体现了宏观系统的无序与微观粒子随机运动的统一性。但在远离平衡态的

[2]第一次对不可逆过程的定量描述可以追溯到傅里叶（Jean Fourier）的热传播定律。

条件下，上述概率的概念则不再成立，即不存在任何普适的定律使得我们能在此条件下推演出系统的“总体”行为。对比稳定系统中的大数定律，其可以清楚地区分均值与噪声，非平衡系统中则可能不同，噪声此时可能不是充当修正均值的角色，而是改变了这些均值。悲观地看，这个基本的概率定理失效了，但换一个角度感受，噪声或者说随机性在宏观层面上仍是主要的。最后，再次引用“分岔”模型，一个个近乎于连续的分岔构成了一个不可逆的演化过程，在每个分岔中的系统，随机性的主导性可以举例为其中一个个体、一种新的思想或行为都能改变全局的宏观状态。

1.1.2 基于人工智能史的研究背景

随着世界上第一台通用计算机于 1946 年在美国诞生，以及图灵（Alan Turing）于 1950 年发表文章《计算机与智能》（Computing Machinery and Intelligence）提出“图灵测试”，“人工智能”（Artificial Intelligence，AI）被公认为缘起 1956 年达特茅斯会议（Dartmouth Conference）[4]。弱 AI（Artificial Narrow Intelligence）随着符号派/逻辑派与统计派/神经网络派两大派系之间的斗争，先后主要经历了基于逻辑的数学定理自动证明的第一次浪潮，基于逻辑与知识的专家系统的第二次浪潮，从存储知识到学习知识的机器学习（Machine Learning）的第三次浪潮。其中，机器学习的定义为一组能够自动从数据中学习知识与模式，并将其应用于预测或是进行决策的方法[5]。模式识别（Pattern Recognition）中一个重要的概念为不确定性，它的出现主要是因为噪声、有限的数据集或时间的推移。基于概率的统计学习（Statistical Learning）则为不确定性的量化与处理提供了一个一致、完备的框架[6]。特征提取（Feature Extraction）也是一个广泛应用统计理论的领域，旨在运用简洁但重要的参数或特征去发现数据中蕴含的“知识”[7]。然而，“没有免费午餐”（No-Free-Lunch）定理陈诉了不存在“全能”的学习器（Universal Leaner）的事实，即每种学习器都存在着各自所擅长的任务，使得机器学习的“自动”性大打折扣。不幸中的万幸，归纳偏置（Inductive Bias）或者说先验知识（Prior Knowledge）的引入使得学习器能在一定程度上避免可预见的“失败”[8]。贝叶斯统计中完备的推断框架包含了先验部分，暗示了基于贝叶斯统计的学习器在某大类任务上的成功。同时，大数据时代背景要求我们自动地从海量、结构化或非结构化的数据中提取“知识”。“回归”则作为一类重要的机器学习任务，承担了连续性数据的预测与分析任务，如：股价预测、气温预测等。

1.1.3 研究意义

给定由 n 个 D 维输入数据（自变量）$\boldsymbol{x}$ 和 1 维输出（因变量）y 组成的数据集 $\mathcal{D}=\{(\boldsymbol{x}_i, y_i)\mid i=1,\ldots,n\}$，回归需要从数据集 $\mathcal{D}$ 中归纳出一个函数 $f:\mathbb{R}^D\to\mathbb{R}$，

使得对于一个新的输入 $\boldsymbol{x}_*$，能够计算出相应的预测值 y_*，特别地，输入可以由 $n\times D$ 的设计矩阵（Design Matrix） $\boldsymbol{X}$ 表示，对应 $n\times 1$ 的输出 $\boldsymbol{y}$。这类任务统称为监督学习（Supervised Learning），本文主要考虑其中的回归任务。许多方法被提出处理该预测任务，此处主要考虑基于限定偏置（Restriction Bias）和基于优选偏置（Preference Bias）的方法。前者通常对预测函数 f 作出了限制，对应于参数模型，如：线性回归、多项式回归。后者则考虑一类可能的函数 f，并且不同的函数具有不同的先验概率，对应于贝叶斯（Bayesian）非参数模型。第一类方法的弊端显而易见，即当输入与输出不满足所限制的函数关系时，预测结果的可信度会十分低，并且，当增加参数以提升模型能力时，也很容易陷入过拟合风险。第二类方法同样存在着严重的问题，因为被考虑的函数类可能包含了无数个函数，那么如何在有限的时间内找到合适的函数？一个答案是，高斯过程（Gaussian Process）可以被用来缓解这个问题。其中，高斯概率分布用来描述随机变量的性质，而随机过程则描述了这一类函数的性质。此时，函数则可以被当作是一串非常长的向量，在一个分量 $\boldsymbol{x}$ 上，确定了一个用于预测的函数值 $f(\boldsymbol{x})$。通过建立在高斯过程上的推断框架，当我们只关注函数在有限个分量上的性质时，该推断的结果等效于我们在过程中已经考虑了所有分量但选择性将其他分量忽略的结果。这便解决了之前的计算性问题，以至于高斯过程回归（Gaussian Process Regression，GPR）[9] 相当具有吸引力。

除了上述最关键的问题以外，高斯过程回归具备着一些鲜明的特征，也在不同方面体现着特别的吸引力。由于是非参数模型，高斯过程回归有能力拟合任意连续函数，以至于我们不用担心模型的拟合能力，虽然说这种拟合能力同样被先验所限制着。并且，高斯过程回归的推断过程完全基于完备的贝叶斯理论，具有坚实的理论基础。描述地说，该推断框架即我们先验地考虑一大类可能的随机函数，而随着一个个数据点被观察到时，我们逐渐拒绝其中不符合数据的那一部分函数。也就是说，预测的不确定性（Uncertainty）也被模型考虑其中。这种不确定性的度量由高斯过程中的方差（Variance）或者说二阶矩（Moment）来体现，而预测从一阶矩向二阶矩甚至高阶矩拓展的过程，是为了更加精确地获取与描述所研究变量分布的必然结果，当然，现实应用中所使用的信息为精度与简单性权衡后的结果。特别地，就算身处于大数据时代，当对于研究问题加以限制时，很有可能出现可用的数据相对很少的情况[5]。并且根据多领域的经验，数据分布会经常显示出一种“长尾（Long-Tailed）”的特征，即那些小概率出现的数据更加重要，举个例子，在搜索引擎上进行搜索时，我们会倾向于使用特定的名词来获取我们想要的信息，而不是大量出现的“的”或者“和”。使用二阶矩信息的模型则在一定程度上考虑了这种问题。图 1.2 则描述了 GPR 简单的推断过程。

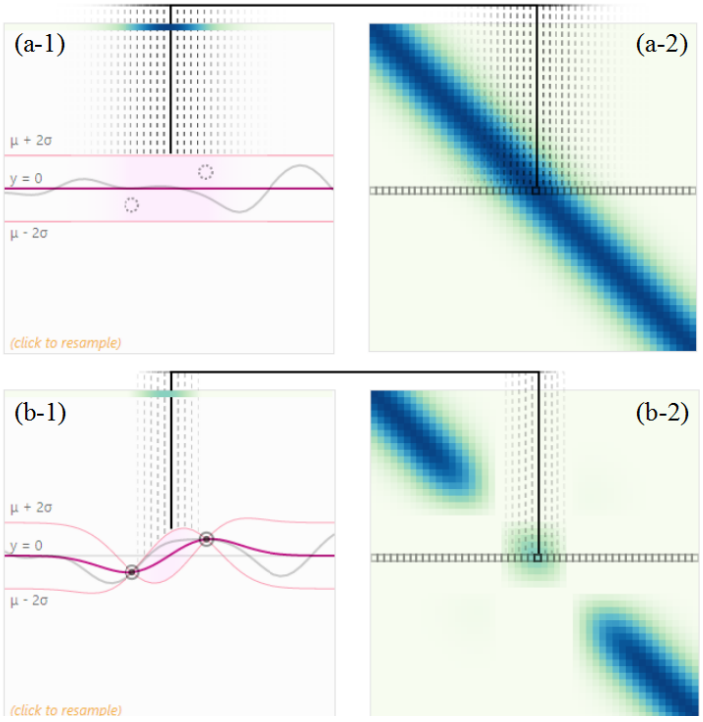

图 1.2　核函数为 SE 核的 GPR 推断示意图。(a)表示先验分布，(b)表示观测到 2 个数据点后的后验分布；(1)中深红色的线表示预测均值，灰色的线表示随机过程的一条样本轨道，浅红色的线为均值加(减)2 倍标准差，表示预测的不确定性；(2)表示相应的核函数，颜色越深方差越大，可以看出 SE 核的 GPR 主要由相邻的函数值影响，并且在有数据观测的地方方差非常小[3]。

1.2 研究内容与创新点

1.2.1 研究内容

本文主要研究高斯过程回归模型，研究的方向主要为该模型在大数据集上的拓展形式，研究框架如图 1.3 所示。根据研究框架，本文除绪论外分为 4 块主要内容，前两部分为综述内容，后两部分则涉及具体模型的改进工作。

首先，对于高斯过程回归在大数据方面改进的综述部分，是必要的。因为 GPR 被应用于机器学习领域约为上世纪 90 年代，在 Rasmussen 出版 GPML[9]后获得了更加广泛的关注，而在该模型提出的时候，就无法避免计算量的问题。在工业界尚未被广泛应用，同样说明该模型总存在局限性，尤其是计算量问题。在这二三十年期间，有众多学者致力于该问题，故不进行一个全面的总结，就难以站在巨人的肩膀上进一步改进，反而浪费精力在 2 次开发上。并且，该领域的综述类文章都存在着局限性[4]，故进一步进行总结也是有必要的。但由于各方面限

[3] 来源于“A Visual Exploration of Gaussian Processes: How to turn a collection of small building blocks into a versatile tool for solving regression problems.”，https://distill.pub/2019/visual-exploration-gaussian-processes/。
[4] 注：作者在完成本文该部分的过程中并未关注 Liu(2020)[74]的工作，因为该文章主题与 Liu(2019)[50]相似，主观认为两者是同一篇。Liu(2020)[74]工作的内容是相对全面的，本文第三章可作为补充材料。

制，该综述部分只考虑应用于 GPR 的方法，如某方法被应用于其他基于核方法的模型中但未应用于 GPR，则本文未加以考虑。并且，该领域的研究还十分活跃，本文不排除存在遗漏部分文献的可能性。

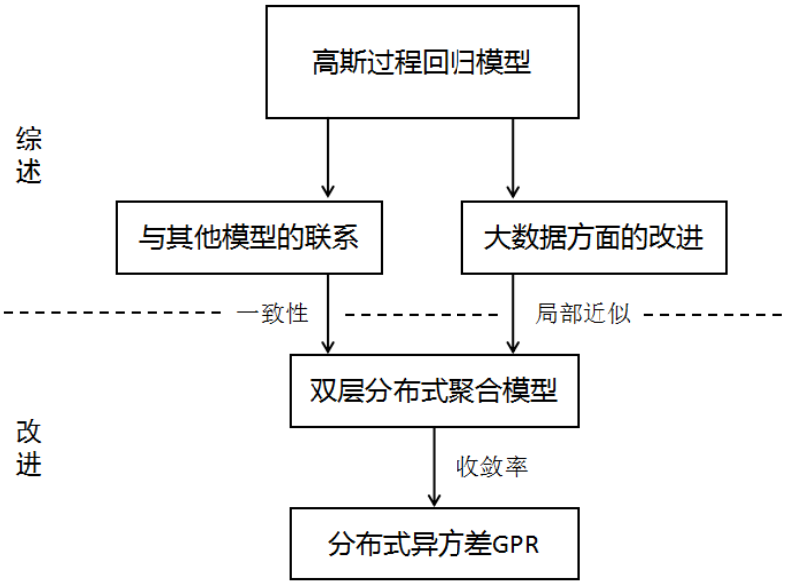

图 1.3　本文研究框架

其次，与其他模型的联系部分也是有需求的。因为在第 4 章使用的“一致性”，出自克里金（Kriging）模型。Kriging 在上世纪 60 年代后被广泛用于地质统计学，其理论性质方面的研究早于 GPR，且在大量的文献中默认 GPR 与 Kriging 是等价的，故本文梳理了一些支持两模型等价的理由，为之后“一致性”的应用作铺垫。此外，GP 与神经网络的联系也是导致 GP 模型在近几年备受关注的一个原因。自从 Neal 于 1995 年的博士论文中阐述了单层无限宽的贝叶斯神经网络有等价的 GP 形式[22]，之后则有大量的研究专注于此，且现在也是非常活跃且极具吸引力的研究方向。再者，添加了部分与其他模型的联系有助于发展 GPR 模型本身，如同为基于核方法的模型，建立了 GPR 与核岭回归之间的联系后，可将核岭回归的部分理论迁移至 GPR 上，但本文不深入考虑类似的迁移过程。

再次，双层分布式聚合模型基于 GPR 的局部近似。该部分内容进一步比较了局部近似的 GPR 模型，并且考虑了预测的一致性，在此基础上导出了该双层分布式的聚合模型。并且，使用了 1 个玩具例子以及 6 个现实数据集，最大的数据量达 3 百万，显示模型在多方面的提升。

最后，对于分布式异方差高斯过程回归模型，本文主要考虑其收敛率，即需要多少个诱导点可以总结所有数据的信息。在该部分，本文分别给出了先验策略以及后验策略，并通过 1 个玩具例子以及 5 个显示数据集验证了通过后验 EM 贪婪算法选择诱导点是有效的。

1.2.2 创新点

本文的创新点可简记如下：

1. 主要总结了 GPR 与 Kriging 模型和神经网络模型的联系；

2. 总结了 GPR 在大数据方面的拓展形式，与 Liu 等（2019）[50]相比，分类更加全面，增加了如全局近似中变分法的细节；

3. 结合了聚合模型中 GPoE[116]注重全局的能力及 GRBCM[120]捕捉局部信息且能保持一致预测的能力，导出了双层聚合模型 GPoGRBCM[147]，其在大数据集上的预测能力更优；

4. 基于 DVSHGP 模型[70]，将同方差 GPR 的收敛率[140]推广到异方差得出选择诱导点个数的先验策略，将贪婪 EM 选择诱导点算法[66]与先验策略结合得到后验策略，可在算法的实际应用中迭代寻找最优的诱导点数。

2 高斯过程回归模型与其他模型的联系

2.1 高斯过程回归简介

作为一个非参数贝叶斯模型，高斯过程回归假设了高斯先验分布来推断隐函数（或真实值函数）$f:\mathbb{R}^D\to\mathbb{R}$，则该隐函数可由均值函数与协方差函数唯一确定：

$$f(\boldsymbol{x})=\mathcal{GP}\left(\mu(\boldsymbol{x}),k(\boldsymbol{x},\boldsymbol{x}')\right) \tag{2.1}$$

其中 $\mu(\boldsymbol{x})$ 是均值函数，通常被设为 0 且不失一般性[5]；$k(\cdot,\cdot)$ 为核函数，不同的核函数使模型能够捕捉不同的统计特征，如周期性、不变点、可加性、对称性等[25]，且核函数中的超参数控制着隐函数的具体形状，如函数是快速变化的或是相对平稳的。于此，本文的实验多使用著名的平方指数核（Squared Exponential，SE）[6]：

$$k(\boldsymbol{x},\boldsymbol{x}')=\sigma_f^2\exp\left(-\frac{1}{2}\sum_{i=1}^{D}\frac{(x_i-x_i')^2}{l_i^2}\right) \tag{2.2}$$

其中 $\boldsymbol{\psi}=\left\{\sigma_f^2,l_1,\cdots,l_D\right\}$ 为该核函数的超参数。

考虑回归任务 $y(\boldsymbol{x})=f(\boldsymbol{x})+\varepsilon$，其中 $\varepsilon\sim\mathcal{N}(0,\sigma_\varepsilon^2)$ 是 *i.i.d.* 噪声，可得似然函数为 $p(y\mid f(\boldsymbol{x}))=\mathcal{N}(f(\boldsymbol{x}),\sigma_\varepsilon^2)$。一般情况，训练标准高斯过程回归模型即是使用训练集 $\mathcal{D}$，通过最大化对数边际似然函数来最优化超参数 $\boldsymbol{\theta}=\left\{\boldsymbol{\psi},\sigma_\varepsilon^2\right\}$，其中对数边际似然函数可以如下表示：

$$\log p(\boldsymbol{y}\mid\boldsymbol{X},\boldsymbol{\theta})=-\frac{n}{2}\log(2\pi)-\frac{1}{2}\left(\boldsymbol{y}(\boldsymbol{K}_{nn}+\sigma_\varepsilon^2\boldsymbol{I})^{-1}\boldsymbol{y}\right)+\log\left|\boldsymbol{K}_{nn}+\sigma_\varepsilon^2\boldsymbol{I}\right| \tag{2.3}$$

其中，$\boldsymbol{K}_{nn}=k(X,X)$ 是 $n\times n$ 的协方差矩阵。此时我们已有训练集 $\mathcal{D}$，及训练完成的超参数 $\boldsymbol{\theta}$，给定测试点 $\boldsymbol{x}_*$，则可由 GP 得出后验分布 $p(f_*\mid\mathcal{D},\boldsymbol{x}_*,\boldsymbol{\theta})$，其均值与方差可以如下表示：

$$\mu_*\left(\boldsymbol{x}_*\right)=\boldsymbol{k}_{n*}^T(\boldsymbol{K}_{nn}+\sigma_\varepsilon^2\boldsymbol{I})^{-1}\boldsymbol{y} \tag{2.4}$$

$$\sigma_*^2\left(\boldsymbol{x}_*\right)=k_{**}-\boldsymbol{k}_{n*}^T(\boldsymbol{K}_{nn}+\sigma_\varepsilon^2\boldsymbol{I})^{-1}\boldsymbol{k}_{n*} \tag{2.5}$$

其中，$k_{**}=k(\boldsymbol{x}_*,\boldsymbol{x}_*)$，$\boldsymbol{k}_{n*}=k(\boldsymbol{X},\boldsymbol{x}_*)$。此时 y_* 的后验分布为 $\mathcal{N}(y_*\mid\mu_*\left(\boldsymbol{x}_*\right),\sigma_*^2\left(\boldsymbol{x}_*\right)+\sigma_\varepsilon^2)$。

2.2 与克里金插值法（Kriging）的联系

令均值函数为常数μ并给定相应的高斯先验 $f(\boldsymbol{x})\sim\mathcal{GP}\left(\mu,k(\boldsymbol{x},\boldsymbol{x}')\right)$，假设真

[5]即模型预测方差不变，预测均值增加了一项均值函数。

[6]同高斯核、径向基函数核（RBF Kernel），且使用自动相关确定（Automatic Relevance Determination, ARD）。

实值y的估计值f不含噪声时，高斯过程回归模型在单个测试点$\boldsymbol{x}_*$上的预测均值与方差分别为：

$$\mu(\boldsymbol{x}_*)=\mu+\boldsymbol{k}_{*n}\boldsymbol{K}_{nn}^{-1}(y-\mu), \tag{2.6}$$

$$\sigma^2(\boldsymbol{x}_*)=k_{**}-\boldsymbol{k}_{*n}\boldsymbol{K}_{nn}^{-1}\boldsymbol{k}_{n*}. \tag{2.7}$$

其中，$\boldsymbol{k}_{*n}=k(\boldsymbol{x}_*,\boldsymbol{X})$，$\boldsymbol{k}_{n*}=k(\boldsymbol{X},\boldsymbol{x}_*)$，$\boldsymbol{K}_{nn}=k(\boldsymbol{X},\boldsymbol{X})$，$k_{**}=k(\boldsymbol{x}_*,\boldsymbol{x}_*)$。

相似地，克里金插值法也将观测值（或预测值）y视为高斯过程上的观测值，为了加以区分，点$\boldsymbol{x}$上的观测值记为$Z(\boldsymbol{x})=\mu+\delta(\boldsymbol{x})$，其中$\delta(\boldsymbol{x})$为零均值的随机过程，其方差函数也使用$C(\boldsymbol{x},\boldsymbol{x}')\equiv\operatorname{cov}\big(Z(\boldsymbol{x}),Z(\boldsymbol{x}')\big)$加以区分。但克里金法以其他观测值的线性估计为基础，该估计是最优线性无偏（Best Linear Unbiased Estimator，BLUE）的[10]。

假设μ是已知的，简单克里金（Simple Kriging）对 y 的估计式如下所示：

$$\sum_{i=1}^{n}\alpha_i Z(\boldsymbol{x}_i)+c \tag{2.8}$$

其中$\boldsymbol{\alpha}^{\top}=[\alpha_1,\cdots,\alpha_n]$和$c$为参数，通过最小化：

$$\mathbb{E}\left(Z(\boldsymbol{x}_*)-\sum_{i=1}^{n}\alpha_i Z(\boldsymbol{x}_i)-c\right)^2 \tag{2.9}$$

可得。此时由最小二乘估计得最优估计参数

$$\boldsymbol{\alpha}^{\top}=\boldsymbol{c}_{*n}\boldsymbol{C}_{nn}^{-1} \tag{2.10}$$

$$c=\left(1-\boldsymbol{c}_{*n}\boldsymbol{C}_{nn}^{-1}\mathbf{1}\right)\mu \tag{2.11}$$

其中$\boldsymbol{c}_{*n}=C(\boldsymbol{x}_*,\boldsymbol{X})$，$\boldsymbol{C}_{nn}=C(\boldsymbol{X},\boldsymbol{X})$，$\mathbf{1}$为$n\times1$维元素全是 1 的向量。因此，简单克里金的最优估计值为

$$Z^*(\boldsymbol{x}_*)=\boldsymbol{c}_{*n}\boldsymbol{C}_{nn}^{-1}\boldsymbol{Z}+\left(1-\boldsymbol{c}_{*n}\boldsymbol{C}_{nn}^{-1}\mathbf{1}\right)\mu \tag{2.12}$$

这里的$\boldsymbol{Z}$等同于数据中的观测值$\boldsymbol{y}$，此时估计的均方误差为

$$\mathbb{E}\left(Z(\boldsymbol{x}_*)-Z^*(\boldsymbol{x}_*)\right)^2=c_{**}-\boldsymbol{c}_{*n}\boldsymbol{C}_{nn}^{-1}\boldsymbol{c}_{n*} \tag{2.13}$$

其中$c_{**}=C(\boldsymbol{x}_*,\boldsymbol{x}_*)$，$\boldsymbol{c}_{n*}=C(\boldsymbol{X},\boldsymbol{x}_*)$。对估计式进行变形，可得

$$Z(\boldsymbol{x}_*)-\mu=\sum_{i=1}^{n}\alpha_i\left(Z(\boldsymbol{x}_i)-\mu\right) \tag{2.14}$$

由$\mathbb{E}\left(Z(\boldsymbol{x}_*)-\mu\right)=\mathbb{E}\left(Z(\boldsymbol{x}_i)-\mu\right)=0$可知，简单克里金的估计在假设下是无偏的。

观察高斯过程回归的预测结果（式（2.6）、（2.7））与简单克里金的预测结果（式（2.12）、（2.13）），可以发现：当核函数$K(\cdot,\cdot)$与协方差函数$C(\cdot,\cdot)$形式相同时，两者的预测结果是相等的。特别地，克里金法考虑的是含噪声的观测值[11]，但这不与高斯过程回归模型中所考虑的无噪声版本相冲突，因为克里金法中的噪声指的是过程噪声，而高斯过程回归模型中的噪声指的是观测噪声。高斯

过程回归模型在实际应用的过程中，其均值函数$\mu(\boldsymbol{x})$经常被先验地令为零，有部分原因是为了保持模型的简洁，因为由预测结果看，均值函数对模型预测的影响只停留于均值上，即其对方差或者说不确定性的度量毫无作用[7]。但是，当考虑到模型训练过程时，结果可能就会变得不一样，因为训练模型时，使用常均值μ或将其强制令为零会导致不同的训练结果，如核函数的超参数不同。一种方式为对数据进行标准化。此外，为了增加模型的拟合或解释能力，一个更加复杂的均值函数$\mu(\boldsymbol{x})$是值得考虑的，只是在实际应用中存在着难以确定均值函数具体形式的问题。学习克里金法对观测值的分解，那么考虑这么一个高斯过程[8]

$$g(\boldsymbol{x}) = f(\boldsymbol{x}) + \boldsymbol{h}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\beta} \tag{2.15}$$

可以使均值函数的确定性问题得到简化，其中$f(\boldsymbol{x}) \sim \mathcal{GP}\left(0, k(\boldsymbol{x}, \boldsymbol{x}')\right)$，$\boldsymbol{h}(\boldsymbol{x})$为一组基函数的集合，$\boldsymbol{\beta}$为额外从数据中确定的参数，且其先验可以假设为$\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{b}, B)$。此外，克里金法也与其他许多模型有着“异曲同工”的联系，如：维纳-柯尔莫可洛夫滤波器（Wiener Kolmogorov Filter）[9]；由再生核希尔伯特空间（Reproducing Kernel Hilbert Space，RKHS）与随机过程的对偶性导出的克里金法与 RKHS 上样条插值的等价性[13]；并且在某些条件下，克里金法可以退化为薄片样条法（Thin-Plate Spline）[13]。

克里金法与高斯过程回归结果的等价性来源于希尔伯特空间上的投影理论（The Projection Theorem）[15]，即 L2 空间上使用条件期望预测等价于最小化均方误差估计[10][16]。根据投影（Projection）的定义，克里金最优预测值$Z^*(\boldsymbol{x}_*)$为随机过程$Z(\boldsymbol{x}_*)$在由$Z(\boldsymbol{x}_i)$张成的空间$span\{Z(\boldsymbol{x}_i), i = 1, \ldots, n\}$上的正交投影。考虑高斯过程回归，根据条件期望的定义，预测值$\mu(\boldsymbol{x}_*) = \mathbb{E}\left(f_* \mid \mathcal{F}\right)$同样为概率空间上的随机变量$f_*$在子空间$\mathcal{L}^2(\Omega, \mathcal{F}, \mathbb{P})$上的正交投影，其中$\mathcal{F}$为由$f_1, \ldots, f_n$生成的$\sigma$代数（$\sigma-$algebra），$f_i = f\left(\boldsymbol{x}_i\right)$。

当$\mu(\boldsymbol{x})$不为常数时，高斯过程回归则与克里金法的预测结果不同。高斯过程回归中，依据条件概率得到的后验结果，均值函数仅从常数变为关于$\boldsymbol{x}$的函数，而基于均值函数为常数假设的简单克里金与普通克里金并不能直接对该问题建模。一般地，采用泛克里金（Universal Kriging）的方法，将均值函数表示为一组基函数的线性组合（等同于式（2.15）），此时，预测与已知$\mu(\boldsymbol{x})$函数形式的高斯过程回归预测结果不一致[10]。

克里金法估计$\delta(\boldsymbol{x})$用来预测$\boldsymbol{x}_*$上的预测值，而维纳滤波器则过滤掉过程噪声$\delta(\boldsymbol{x})$使得序列显示出原有的趋势$\mu(\boldsymbol{x})$。相比于克里金插值法，只考虑单个位

[7] Kriging 版本见 Chiles 和 Delfiner（2012）[12]。
[8]拓展内容可见 Rasmussen 和 Williams（2006）[9]的第 2.7 章，并且该形式与泛克里金相同。
[9]克里金中滤波或平滑的部分可参考 Chiles 和 Delfiner（2012）[12]。
[10]条件期望的最优估计性对应其他损失函数的情况可参考 Banerjee 等（2005）[18]。

置 $\boldsymbol{x}$ 上的单个观测值，高斯过程回归则在此基础上，考虑多个观测值，使得其在插值法的基础上，可以估计出观测值的观测噪声。同样地，当克里金包含观测噪声时，回归克里金（Regressing Kriging）与高斯过程回归等价[19]。一般地，克里金法是为了处理那些不可重复观测的、单次测量需要花费大量资源的数据而提出的，因而在应用中普遍地做法为假设均值 $\mu(\boldsymbol{x})$ 未知，进而使用基于数据估计均值的普通克里金法。但是当数据量非常大时，使用基于已知均值假设的简单克里金法或说高斯过程回归则是合理的，如观测是基于时空的动态过程，或在截面时刻有大量不同位置的观测数据[12]。特别地，鉴于高斯过程回归与克里金法的应用领域不同，两者对协方差函数的处理也略有不同。在实际应用中，克里金法发展了估计变异函数（Variogram）$\gamma(h)=0.5\mathbb{V}\mathrm{ar}[Z(x+h)-Z(x)]$ 的方法，用来度量空间中不同点的基于距离的相关性。而高斯过程回归则致力于在复杂的参数空间上建立合适的模型，则使用了不同种类的核函数。同样地，依据不同的目的，克里金法考虑如何适应空间建模过程，如在非欧式空间上建模；而高斯过程回归则倾向于提升其处理大数据的能力，如分布式高斯过程回归，但两者对复杂过程的建模方向存在着共通的部分，如两者的研究方向都包含非平稳过程或随机场的建模、模型对于数据的异方差性的处理等。

普通克里金法同样也为线性估计，有 $\sum_{i=1}^{n}\alpha_i Z(\boldsymbol{x}_i)$。因为无偏估计，则 $\sum_{i=1}^{n}\alpha_i=1$。同样最小化均方预测误差，使用拉格朗日乘子法（Language Multipliers），可得最优的参数值：

$$\boldsymbol{\alpha}^{\top}=\left[\boldsymbol{c}_{n*}+\left(1-\boldsymbol{c}_{*n}\boldsymbol{C}_{nn}^{-1}\mathbf{1}\right)\left(\mathbf{1}^{\top}\boldsymbol{C}_{nn}^{-1}\mathbf{1}\right)^{-1}\mathbf{1}\right]^{\top}\boldsymbol{C}_{nn}^{-1}。\tag{2.16}$$

因此，预测的均值与方差可以如下表示：

$$Z^{*}(\boldsymbol{x}_{*})=\boldsymbol{c}_{*n}\boldsymbol{C}_{nn}^{-1}\boldsymbol{y}+\left(1-\boldsymbol{c}_{*n}\boldsymbol{C}_{nn}^{-1}\mathbf{1}\right)\left(\mathbf{1}^{\top}\boldsymbol{C}_{nn}^{-1}\mathbf{1}\right)^{-1}\left(\mathbf{1}^{\top}\boldsymbol{C}_{nn}^{-1}\mathbf{Z}\right)\tag{2.17}$$

$$\mathrm{E}\left(Z(\boldsymbol{x}_{*})-Z^{*}(\boldsymbol{x}_{*})\right)^{2}=c_{**}-\boldsymbol{c}_{*n}\boldsymbol{C}_{nn}^{-1}\boldsymbol{c}_{n*}+\left(1-\boldsymbol{c}_{*n}\boldsymbol{C}_{nn}^{-1}\mathbf{1}\right)^{2}\left(\mathbf{1}^{\top}\boldsymbol{C}_{nn}^{-1}\mathbf{1}\right)^{-1}\tag{2.18}$$

其中，常数均值的估计值与估计方差分别为：

$$\hat{\mu}=\left(\mathbf{1}^{\top}\boldsymbol{C}_{nn}^{-1}\mathbf{1}\right)^{-1}\left(\mathbf{1}^{\top}\boldsymbol{C}_{nn}^{-1}\mathbf{Z}\right)\tag{2.19}$$

$$\mathbb{V}\mathrm{ar}=\left(\mathbf{1}^{\top}\boldsymbol{C}_{nn}^{-1}\mathbf{1}\right)^{-1}\tag{2.20}$$

可以看出，当均值已知，或者估计方差非常小时，普通克里金与简单克里金等价。且一般而言，相对预测值的预测误差而言，均值的估计误差一般非常小[19]。

克里金法存在着一个吸引人的大样本性质——逐点一致性（Pointwise Consistency）[20]，希望该性质能被移植到高斯过程回归上，因为比起地质统计学的方法，机器学习方法应用的领域更容易获得大量的数据。简单地，如果点 $\boldsymbol{x}_{*}$ 附着于输入集 $\{\boldsymbol{x}_n, n\geq 1\}$，因为核函数 $k(\cdot,\cdot)$ 是连续的，所以期望的预测误差 $\sigma^{2}(\boldsymbol{x}_{*})$

将会趋于 0，即 $\lim_{n\to\infty}\sigma^2(\boldsymbol{x}_*)=0$。此时，随机变量的估计具有逐点一致性，因为估计是无偏的，随机变量的预测 $\hat{Z}(x_*)$ 收敛到真实的随机变量 $Z(x_*)$。特别地，克里金法将隐函数 f 视为随机过程 Z 的一个样本轨道（Sample Path），在现实应用中，观测值 $z(\boldsymbol{x})=Z(\omega,\boldsymbol{x})$ 可以被视为一个样本轨道 $Z(\omega,\cdot)$ 在 $\boldsymbol{x}$ 上的观测或 $\boldsymbol{x}$ 上随机变量 $Z(\cdot,\boldsymbol{x})$ 的一个采样，其中 $\omega\in\Omega$。所以除了随机变量预测的一致性，逐点一致性是否对于所有样本轨道 $Z(\omega,\cdot)$ 成立则更进一步影响预测的精确性，换句话说，当该性质对所有样本轨道成立时，只要在现实应用中数据量 $n\to\infty$，那么克里金法或者与之形式相对应的高斯过程回归的预测值 $\hat{Z}(\omega,x_*)$ 能够收敛到真实值 $Z(\omega,x_*)$。以下给出部分结论，细节见 Vazquez 和 Bect（2009）[20]。当随机过程 Z 不为高斯过程时，对任意样本轨道 $Z(\omega,\cdot)\in\mathcal{H}$，逐点一致性成立 $\lim_{n\to\infty}\hat{Z}(\omega,\boldsymbol{x}_*)=Z(\omega,\boldsymbol{x}_*)$。特别地，当 Z 为高斯过程时，给定核函数 $k(\cdot,\cdot)$，存在一个函数的集合 $\mathcal{G}$，使得对任意样本轨道 $Z(\omega,\cdot)\in\mathcal{G}$，逐点一致性成立。集合 $\mathcal{G}$ 存在，需要一些特殊的条件，Vazquez 和 Bect（2009）[20]的定理 2 给出了一个可行的条件，一是勒贝格常数（Lebesgue Constant）是有界的，二是核函数 $k(\cdot,\cdot)$ 满足文献中的假设 2。特别地，常用的高斯核函数（同称为径向基（Radial Basis）核、平方指数（Square Exponential）核）不满足该条件，但满足的核函数中包括一些其他常用的指数核与马顿（Matérn）核。

另外的大样本性质，考虑包含观测噪声的高斯过程回归，可将实际的观测建模为 $Z(\boldsymbol{x})=\mu+\delta(\boldsymbol{x})+\varepsilon(\boldsymbol{x})$，其中 ε 为噪声有 $\varepsilon(\boldsymbol{x})\sim\mathcal{N}(0,\sigma_\varepsilon^2)$。此时的预测均值与方差为

$$\mu(\boldsymbol{x}_*)=\mu+\boldsymbol{k}_{*n}\left(\boldsymbol{K}_{nn}+\sigma_\varepsilon^2\boldsymbol{I}\right)^{-1}(y-\mu), \tag{2.21}$$

$$\sigma^2(\boldsymbol{x}_*)=k_{**}-\boldsymbol{k}_{*n}\left(\boldsymbol{K}+\sigma_\varepsilon^2\boldsymbol{I}\right)^{-1}\boldsymbol{k}_{n*}+\sigma_\varepsilon^2. \tag{2.22}$$

当 $n\to\infty$ 时，有预测 $\mu(\boldsymbol{x}_*)\to Z(x_*)$，$\sigma^2(\boldsymbol{x}_*)\to\sigma_\varepsilon^2$ [21]。

2.3 与神经网络的联系

高斯过程回归的研究在近 20 年内的关注度逐渐提高，其中一个重要的原因是它与神经网络的联系。早在 1995 年 Neal 就阐述了单层无限宽的神经网络与高斯过程的等价性[22]，近几年也有原来越多的理论研究建立在两个模型的联系之上。

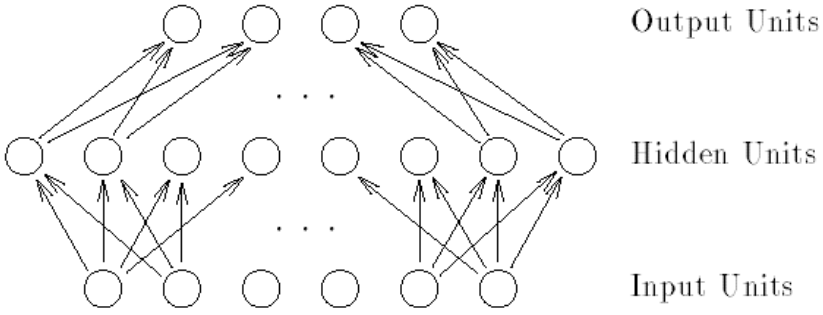

图 2.1　含单隐藏层神经网络模型的简单示意图[22]

如图 2.1，考虑一个含单隐藏层的神经网络模型，输入 $\boldsymbol{x}$ 有 I 个特征 x_i，中间隐藏层 H 的第 j 个神经元的值为 $h_j(\boldsymbol{x})$，第 k 个输出 y_k 的值由函数 $f_k(\boldsymbol{x})$ 确定，有：

$$f_k(\boldsymbol{x}) = b_k + \sum_{j=1}^{H} v_{jk} h_j(\boldsymbol{x}) \tag{2.23}$$

$$h_j(\boldsymbol{x}) = \tanh\left(a_j + \sum_{i=1}^{I} u_{ij} x_i \right) \tag{2.24}$$

其中 u_{ij} 为第 i 个输入特征到第 j 个隐藏层神经元的权重，a_j 为第 j 个隐藏层神经元的偏置，$\tanh(z) = (e^z - e^{-z})/(e^z + e^{-z})$ 为双曲正切激活函数，v_{jk} 为第 j 个隐藏层神经元到第 k 个输出的权重，b_k 为第 k 个输出的偏置。

首先假设相互独立的 v_{jk} 与 b_k 的先验都为零均值的高斯分布，它们的方差分别为 σ_v^2 和 σ_b^2。由独立性可得到第 j 个隐藏层神经元对每个输出的期望贡献为 $\mathbb{E}\left[v_{jk} h_j(\boldsymbol{x})\right] = \mathbb{E}\left[v_{jk}\right]\mathbb{E}\left[h_j(\boldsymbol{x})\right] = 0$，且贡献方差为 $\mathbb{E}\left[v_{jk} h_j(\boldsymbol{x})\right]^2 = \mathbb{E}\left[v_{jk}\right]^2 \mathbb{E}\left[h_j(\boldsymbol{x})\right]^2 = \sigma_v^2 \mathbb{E}\left[h_j(\boldsymbol{x})\right]^2$。假设所有隐藏层神经元 $h_j(\boldsymbol{x})$ 有相同的方差 $V(\boldsymbol{x}) = \mathbb{E}\left[h_j(\boldsymbol{x})\right]^2$，由中心极限定理（Central Limit Theorem）可得，当神经元个数 H 非常大时，所有神经元贡献的总和服从方差为 $H\sigma_v^2 V(\boldsymbol{x})$ 的高斯分布，加上偏置 b_k 后 $f_k(\boldsymbol{x})$ 的先验分布依旧是高斯的，其方差为 $H\sigma_v^2 V(\boldsymbol{x}) + \sigma_b^2$。为避免方差不存在，可令 $\sigma_v = c_v H^{-1/2}$，其中 c_v 为固定的常数。

进一步，有 n 个输入 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$，考虑第 k 个输出 $f_k(\boldsymbol{x}_1), \ldots, f_k(\boldsymbol{x}_n)$ 的联合分布，有协方差：

$$\begin{aligned} \mathbb{E}\left[f(\boldsymbol{x}_p) f(\boldsymbol{x}_q) \right] &= \sigma_b^2 + \sum\nolimits_j \sigma_v^2 \mathbb{E}\left[h_j(\boldsymbol{x}_p) h_j(\boldsymbol{x}_q) \right] \\ &= \sigma_b^2 + c_v k\left(\boldsymbol{x}_p, \boldsymbol{x}_q \right) \end{aligned} \tag{2.25}$$

其中 $k\left(\boldsymbol{x}_p, \boldsymbol{x}_q\right) = \mathbb{E}\left[h_j(\boldsymbol{x}_p) h_j(\boldsymbol{x}_q) \right]$ 对于所有中间层神经元成立，即可将该先验 f_k 看作一个高斯过程。此时，当 $k\left(\boldsymbol{x}_p, \boldsymbol{x}_q\right)$ 已知或被估计后，将 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1}$ 看作训练集的输入，$\boldsymbol{x}_n$ 为测试集的输入，由联合分布从 $f_k(\boldsymbol{x}_1), \ldots, f_k(\boldsymbol{x}_{n-1})$ 推断 $f_k(\boldsymbol{x}_n)$ 分布的过程等同于高斯过程回归的推断过程。对于不同的第 $k1$ 和 $k2$ 个输出，由权重的先验独立性假设可得 $f_{k1}(\boldsymbol{x}_p)$ 和 $f_{k2}(\boldsymbol{x}_q)$ 的协方差为零，即当神经网络中间层的

神经元个数趋于无穷时，不同的输出对于网络的训练不会相互影响，换句话说，训练一个有 k 维输出的神经网络等价于训练 k 个只有 1 维输出的神经网络。这反过来也可以用于理解：为什么一般的高斯过程回归模型只有单个个输出。特别地，该结论对输入层到隐藏层间的参数不需要假设，且激活函数只需要是有界的，隐藏层到输出层的参数也并不需要高斯分布的假设，只需要假设分布满足均值为零方差有限。一个具有更弱条件的证明可以见 Hanin（2021）[23]。

更进一步，深度神经网络同样有等价的高斯过程形式。考虑 L 层的全链接深度神经网络，宽度为 N，非线性激活函数记为 ϕ，其中模型输入记为 $\boldsymbol{x}^0 \equiv \boldsymbol{x}$，激活前和激活后的第 j 个分量可以分别表示为 z_j^l 和 x_j^l。假设 z_j^{l-1} 关于 j 是独立并且服从相同的高斯过程，此时 $x_j^l(x)$ 也是独立同分布的。考虑第 l 层网络：

$$x_j^l(\boldsymbol{x}) = \phi(z_j^{l-1}(\boldsymbol{x})) \tag{2.26}$$

$$z_i^l(\boldsymbol{x}) = b_i^l + \sum\nolimits_{j=1}^{N_l} W_{ij}^l x_j^l(\boldsymbol{x}) \tag{2.27}$$

可以看出，$z_i^l(\boldsymbol{x})$ 是许多随机变量的线性组合，当 $N_l \to \infty$ 时，根据多元中心极限定理，任何由 n 个输入决定的有限个随机变量的集合 $\left\{z_i^l(\boldsymbol{x}_1),\ldots,z_i^l(\boldsymbol{x}_n)\right\}$ 服从联合高斯分布，或者用高斯过程来表示 $z_i^l(\boldsymbol{X}) \sim \mathcal{GP}\left(\boldsymbol{0}, \boldsymbol{K}^l\right)$。此处两两之间的协方差可记为：

$$K^l = k^l\left(\boldsymbol{x},\boldsymbol{x}'\right) \equiv \mathbb{E}(z_i^l(\boldsymbol{x}) z_i^l(\boldsymbol{x}')) = \sigma_b^2 + \sigma_w^2 \mathbb{E}_{z_i^{l-1}\sim\mathcal{GP}\left(0,K^{l-1}\right)}\left[\phi\left(z_i^{l-1}(\boldsymbol{x})\right)\phi\left(z_i^{l-1}(\boldsymbol{x}')\right)\right] \tag{2.28}$$

其中，对于 GP 求期望等价于在 $z_i^{l-1}(\boldsymbol{x})$ 和 $z_i^{l-1}(\boldsymbol{x}')$ 的联合分布上求积分。特别地，该部分内容可以用零均值、协方差矩阵与 $k^{l-1}\left(\boldsymbol{x},\boldsymbol{x}'\right)$、$k^{l-1}\left(\boldsymbol{x},\boldsymbol{x}\right)$ 和 $k^{l-1}\left(\boldsymbol{x}',\boldsymbol{x}'\right)$ 有关的高斯分布来表示。故上式同样可以由不同层元素的协方差关系表示如下：

$$K^l = k^l\left(\boldsymbol{x},\boldsymbol{x}'\right) = \sigma_b^2 + \sigma_w^2 F_\phi\left(k^{l-1}\left(\boldsymbol{x},\boldsymbol{x}'\right), k^{l-1}\left(\boldsymbol{x},\boldsymbol{x}\right), k^{l-1}\left(\boldsymbol{x}',\boldsymbol{x}'\right)\right) \tag{2.29}$$

其中，确定性函数 F 的形式只取决于激活函数 ϕ 的形式。对于广义的激活函数 ϕ，积分需要通过数值方法求解，具体可见 Lee 等（2018）[24]。而对于一些具体的激活函数，可以求得上式的解析形式，如 ReLu 激活函数 $\phi(x) = \max(0,x)$，有

$$K^l = k^l\left(\boldsymbol{x},\boldsymbol{x}'\right) = \sigma_b^2 + \frac{\sigma_w^2}{2\pi}\sqrt{k^{l-1}\left(\boldsymbol{x},\boldsymbol{x}\right)k^{l-1}\left(\boldsymbol{x}',\boldsymbol{x}'\right)}\left(\sin\theta_{x,x'}^{l-1} + \left(\pi - \theta_{x,x'}^{l-1}\right)\cos\theta_{x,x'}^{l-1}\right), \tag{2.30}$$

$$\theta_{x,x'}^l = \cos^{-1}\left(\frac{k^l\left(\boldsymbol{x},\boldsymbol{x}'\right)}{\sqrt{k^l\left(\boldsymbol{x},\boldsymbol{x}\right)k^l\left(\boldsymbol{x}',\boldsymbol{x}'\right)}}\right) \tag{2.31}$$

其中 θ 为 $\boldsymbol{x}$ 和 $\boldsymbol{x}'$ 所呈的夹角度数。另一个激活函数的例子见 Williams（1997）[25] 的附录 7.4。

假设 $W_{ij}^0 \sim \mathcal{N}\left(0,\sigma_w^2/d_{in}\right)$，$b_j^0 \sim \mathcal{N}\left(0,\sigma_b^2\right)$，可以得到初始协方差

$$K^0 = k^0(\boldsymbol{x},\boldsymbol{x}') = \mathbb{E}\left[z_j^0(\boldsymbol{x})z_j^0(\boldsymbol{x}')\right] = \sigma_b^2 + \sigma_w^2\left(\boldsymbol{x}\cdot\boldsymbol{x}'/d_{in}\right) \quad (2.32)$$

通过 L 层迭代，可以求得第 L 层的核函数 $K^L = k^L(\boldsymbol{x},\boldsymbol{x}')$，以该核函数作为协方差函数的 GP 称为神经网络高斯过程（neural network Gaussian process，NNGP）[24]。对于预测输入 $\boldsymbol{x}_*$，有预测均值与方差

$$\mathbb{E}(y_*) = k^L(x_*,\boldsymbol{X})\left(k^L(\boldsymbol{X},\boldsymbol{X}) + \sigma_\varepsilon^2\boldsymbol{I}\right)^{-1}\boldsymbol{y} \quad (2.33)$$

$$\mathbb{V}\mathrm{ar}(y_*) = k^L(x_*,x_*) - k^L(x_*,\boldsymbol{X})\left(k^L(\boldsymbol{X},\boldsymbol{X}) + \sigma_\varepsilon^2\boldsymbol{I}\right)^{-1}k^L(\boldsymbol{X},x_*) + \sigma_\varepsilon^2 \quad (2.34)$$

Cho 和 Saul（2009）[26]建议使用递归核函数（Recursive Kernel）去模拟一个更大、更深层的神经网络计算。通常核函数可以表示为 $k(\boldsymbol{x},\boldsymbol{x}') = \Phi(\boldsymbol{x})\cdot\Phi(\boldsymbol{x}')$，其中 $\Phi(\boldsymbol{x})$ 为特征映射，而递归核则使用新的特征映射，如 $\Phi\left(\Phi(\boldsymbol{x})\right)$。然而该构造核函数的方法并不符合核函数的创建规则，Matthew 等（2018）[27]则使用了一个特别的特征映射 $\phi(x) = \Theta(x)x^r$ 处理该问题，其中以一维输入举例，$r = 0,1,2,3$，Θ 为单位跃阶函数（Heaviside step function）。

一部分关于 NNGP 的研究在于拓展输入噪声，如将 $y(\boldsymbol{x}) = f(\boldsymbol{x}) + \varepsilon$ 拓展为有输入误差的模型 $y(\boldsymbol{x}) = f(\boldsymbol{x}+\boldsymbol{u}) + \varepsilon$ [28]。假设 $f \sim \mathcal{NNGP}\left(0, k^L(\boldsymbol{x},\boldsymbol{x}')\right)$，则包含输入误差的输出 y 的协方差函数可以表示为

$$\begin{aligned}&\mathbb{C}\mathrm{ov}\left[y(\boldsymbol{x}), y(\boldsymbol{x}')\right] = \mathbb{C}\mathrm{ov}\left[f(\boldsymbol{x}+\boldsymbol{u}), f(\boldsymbol{x}'+\boldsymbol{v})\right] = \\ &\mathbb{E}_{u,v}\left[k^L\left(\boldsymbol{x}+\boldsymbol{u},\boldsymbol{x}'+\boldsymbol{v}\right)\right] = \iint_{u,v} k^L\left(\boldsymbol{x}+\boldsymbol{u},\boldsymbol{x}'+\boldsymbol{v}\right)p_u(\boldsymbol{u})p_v(\boldsymbol{v}) \\ &\triangleq C^L\left(\boldsymbol{x},\boldsymbol{x}'\right)\end{aligned} \quad (2.35)$$

对于预测点 x_*，同样可得修正的核函数

$$\begin{aligned}&C^L\left(\boldsymbol{x}_*,\boldsymbol{x}\right) = \mathbb{C}\mathrm{ov}\left[f(\boldsymbol{x}_*), y(\boldsymbol{x})\right] \\ &= \mathbb{E}_u\left[k^L\left(\boldsymbol{x}_*,\boldsymbol{x}+\boldsymbol{u}\right)\right] = \int_u K^L\left(\boldsymbol{x}_*,\boldsymbol{x}+\boldsymbol{u}\right)p_u(\boldsymbol{u})\end{aligned} \quad (2.36)$$

考虑输入误差的 NNGP 模型的训练与预测与 GPR、NNGP 类似，训练使用了对数伪似然函数

$$L(\boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{y}^\top\left(C^L(\boldsymbol{X},\boldsymbol{X}) + \sigma_\varepsilon^2\boldsymbol{I}\right)^{-1}\boldsymbol{y} - \frac{1}{2}\log\left|C^L(\boldsymbol{X},\boldsymbol{X}) + \sigma_\varepsilon^2\boldsymbol{I}\right| + constant \quad (2.37)$$

预测为

$$\mathbb{E}(y_*) = C^L(\boldsymbol{x}_*,\boldsymbol{X})\left(C^L(\boldsymbol{X},\boldsymbol{X}) + \sigma_\varepsilon^2\boldsymbol{I}\right)^{-1}\boldsymbol{y} \quad (2.38)$$

$$\mathbb{V}\mathrm{ar}(y_*) = C^L(\boldsymbol{x}_*,\boldsymbol{x}_*) - C^L(\boldsymbol{x}_*,\boldsymbol{X})\left(C^L(\boldsymbol{X},\boldsymbol{X}) + \sigma_\varepsilon^2\boldsymbol{I}\right)^{-1}C^L(\boldsymbol{X},\boldsymbol{x}_*) + \sigma_\varepsilon^2 \quad (2.39)$$

然而，其中的修正核函数 $C^L\left(\boldsymbol{x},\boldsymbol{x}'\right)$ 无法解析，需要通过近似方法估计，如蒙特卡洛（Monte-Carlo）近似，具体的近似过程见 Lee 等（2020）[28]。而使用伪似然对数作为训练的目标函数也是为了避免过度使用近似方法。特别地，Lee 等（2020）[28]中的命题 5.1 与命题 5.2 为修正 NNGP 的预测提供了优良的预测性质，

即预测是 BLUE 的，并且比未修正的模型更加有效。

对于回归模型的噪声拓展也可体现在神经网络的预测中[29]，如假设神经网络的模型为

$$\boldsymbol{z}^l = \boldsymbol{W}^l\left(\boldsymbol{x}^{l-1}\odot\boldsymbol{u}^{l-1}\right)+\boldsymbol{b}^l,\ \ l=1,\ldots,L \tag{2.40}$$

其中$\odot$可以是加法或者乘法，$\boldsymbol{u}$为输入噪声向量。则第l层的$nN_l\times nN_l$的协方差矩阵可以写成

$$\boldsymbol{K}^l = k^l(\boldsymbol{X},\boldsymbol{X})\otimes\boldsymbol{I}_{N_l} \tag{2.41}$$

其中$\otimes$为克罗内克积（Kronecker Product）。确切地考虑协方差矩阵$\boldsymbol{K}^l$中的各个元素$k^l_{ds}(\boldsymbol{x}_i,\boldsymbol{x}_j)$，其中个第 d 个和 s 个输出$d,s\in\{1,\ldots,N_l\}$以及第 i 和第 j 个输入$i,j\in\{1,\ldots,n\}$，有：

$$\begin{aligned} k^l_{ds}(\boldsymbol{x}_i,\boldsymbol{x}_j) &= k^l(\boldsymbol{x}_i,\boldsymbol{x}_j)\otimes\boldsymbol{I}_{ds} \\ &= \begin{cases} \mathbb{E}\left[z^l_d(\boldsymbol{x}_i)z^l_s(\boldsymbol{x}_j)\right], d=s \\ 0, else \end{cases} \end{aligned} \tag{2.42}$$

考虑 d=s 时，有

$$k^l_d(\boldsymbol{x}_i,\boldsymbol{x}_j) = \begin{cases} \sigma_w^2\mathbb{E}_{d'}\left[\phi\left(z^{l-1}_{d'}(\boldsymbol{x}_i)\right)\phi\left(z^{l-1}_{d'}(\boldsymbol{x}_j)\right)\right], i\neq j \\ \sigma_w^2\left\{\mathbb{E}_{d'}\left[\phi\left(z^{l-1}_{d'}(\boldsymbol{x}_i)\right)^2\right]\odot\mu_2\right\}, i=j \end{cases} \tag{2.43}$$

其中μ_2为输入噪声的二阶矩。给定初始条件$k^0\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right)=\mathbb{E}_u\left[\left(\boldsymbol{x}_i\odot\boldsymbol{u}\right)\left(\boldsymbol{x}_j\odot\boldsymbol{u}\right)\right]$并使每层神经网络的宽度$N_1,\ldots,N_L$趋于无穷，迭代结构使得 GPNN 给出$\boldsymbol{f}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{K}^L)$。此时同 GPR 一样，可以得到在$\boldsymbol{x}_*$上的预测结果

$$p\left(y_*\mid\boldsymbol{f},\boldsymbol{x}_*,\boldsymbol{y}\right)=\mathcal{N}\left(\boldsymbol{k}^\top\left(\boldsymbol{K}^L+\sigma_\varepsilon^2\boldsymbol{I}\right)^{-1}\boldsymbol{y},k^L\left(\boldsymbol{x}_*,\boldsymbol{x}_*\right)-\boldsymbol{k}^\top\left(\boldsymbol{K}^L+\sigma_\varepsilon^2\boldsymbol{I}\right)^{-1}\boldsymbol{k}+\sigma_\varepsilon^2\right) \tag{2.44}$$

其中$\boldsymbol{k}=\left[k_1^L\left(\boldsymbol{x}_1,\boldsymbol{x}_*\right),\ldots,k_1^L\left(\boldsymbol{x}_n,\boldsymbol{x}_*\right),\ldots,k_{N_L}^L\left(\boldsymbol{x}_n,\boldsymbol{x}_*\right)\right]^\top$，一般考虑$N_L=1$。

还有一部分关于 NNGP 的研究在于拓展神经网络的结构，如：卷积神经网络（Convolutional Neural Networks，CNN）、残差神经网络（Residual Network，ResNet）是否与 GP 存在某种联系？当卷积核数量趋于无穷时，CNN 同 NN 一样，与 GP 存在着等价性[30]。

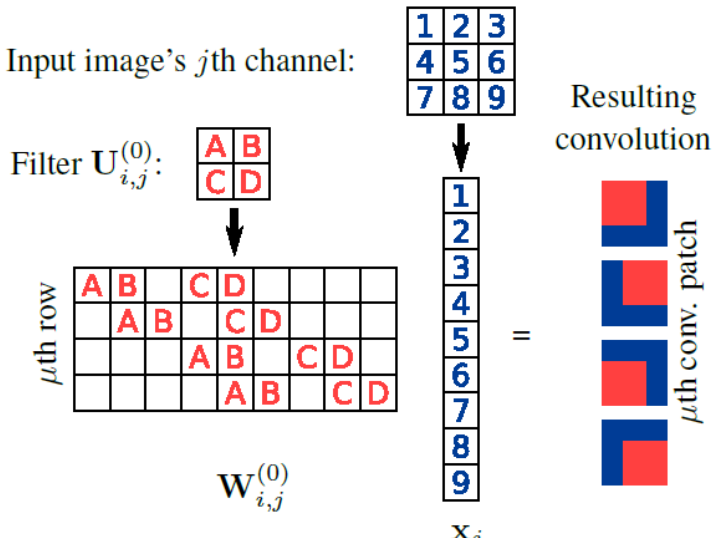

图 2.2 卷积神经网络的矩阵形式[30]。第 j 个通道的矩阵拉长为一个向量 $\boldsymbol{x}_j$，2 维的卷积核 $\boldsymbol{U}_{i,j}^{(0)}$ 应用于第 j 个通道可以表示为矩阵乘法 $W_{i,j}^{(0)}\boldsymbol{x}_j$，其中 $W_{i,j}^{(0)}$ 中空白表示值为零。第 μ 行的 $W_{i,j}^{(0)}$ 对应于卷积核扫描通道 j 的第 μ 块区域（μth-patch）。

给定长为 $H^{(0)}$、宽为 $D^{(0)}$、通道数为 $C^{(0)}$ 的图片输入 $\boldsymbol{X}$（如：RGB 图片的通道数为 3），矩阵的形状可以表示为 $C^{(0)}\times\left(H^{(0)}D^{(0)}\right)$，其中每行的分量可以表示为 $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{C^{(0)}}$，有 $\boldsymbol{x}_j=\left[x_{1,1,j},\ldots,x_{1,D^{(0)},j},\ldots,x_{H^{(0)},D^{(0)},j}\right]$。给定第 l 层的卷积核数量为 $C^{(l)}$，第 i 个卷积核得到的未激活的结果可如下表示：

$$\boldsymbol{a}_i^{(l)}(\boldsymbol{X}):=b_i^{(l)}\mathbf{1}+\sum_{j=1}^{C^{(l)}}\boldsymbol{W}_{i,j}^{(l)}\phi\left(a_j^{(l-1)}(\boldsymbol{X})\right) \tag{2.45}$$

其中 $W_{i,j}^{(l)}$ 为权重矩阵，其形式可见图 2.2。特别地，当 $l=1$ 时，我们令 $\phi\left(a_j^{(l-1)}(\boldsymbol{X})\right)=\boldsymbol{x}_j$。

将向量 $\boldsymbol{a}_i^{(l)}(\boldsymbol{X})$ 按行拼接成 $C^{(l)}\times\left(H^{(l)}D^{(l)}\right)$ 的矩阵 $\boldsymbol{A}^{(l)}(X)$，在回归任务中，最后一层（L+1）通常是全连接层（Fully-Connected Layer），只需令 $H^{(L+1)}=D^{(L+1)}=1$。假设

$$U_{i,j,x,y}^{(l)}\sim\mathcal{N}\left(0,\sigma_w^2/C^{(l)}\right),b_i^{(l)}\sim\mathcal{N}\left(0,\sigma_b^2\right) \tag{2.46}$$

考虑两个输入 $\boldsymbol{X}$ 和 $\boldsymbol{X}'$，有

$$\boldsymbol{a}_i^{(l)}(\boldsymbol{X},\boldsymbol{X}'):=\begin{pmatrix}\boldsymbol{a}_i^{(l)}(\boldsymbol{X})\\ \boldsymbol{a}_i^{(l)}(\boldsymbol{X}')\end{pmatrix}=b_i^{(l)}\mathbf{1}+\sum_{j=1}^{C^{(l)}}\begin{pmatrix}\boldsymbol{W}_{i,j}^{(l)} & 0\\ 0 & \boldsymbol{W}_{i,j}^{(l)}\end{pmatrix}\phi\left(\boldsymbol{a}_j^{(l-1)}(\boldsymbol{X},\boldsymbol{X}')\right) \tag{2.47}$$

特别地，当 $l=1$ 时令 $\phi\left(\boldsymbol{a}_j^{(l-1)}(\boldsymbol{X},\boldsymbol{X}')\right)=\left(\boldsymbol{x}_i,\boldsymbol{x}_i'\right)^\top$。当 $C^{(l-1)}\to\infty$ 时，由多元中心极限定理得，$\boldsymbol{A}^{(l)}(\boldsymbol{X},\boldsymbol{X}')$ 中的所有元素联合服从多元高斯分布。

此时，即可考虑 CNN 等价的核函数的具体形式，因为如上所述要推断出 $\boldsymbol{a}_i^{(l)}(\boldsymbol{X})$ 和 $\boldsymbol{a}_i^{(l)}(\boldsymbol{X}')$ 协方差是非常复杂的，而在回归任务中，我们通常只关心某个

输出的方差而不是考虑所有输出间的关系，即 $\mathrm{diag}\left(\mathbb{C}\mathrm{ov}\left[\boldsymbol{a}_i^{(l)}(\boldsymbol{X}),\boldsymbol{a}_i^{(l)}(\boldsymbol{X}')\right]\right)$。为了简单起见，首先重新将模型改写如下：

$$A_{i,u}^{(l+1)}(\boldsymbol{X})=b_i^{(l+1)}+\sum_{j=1}^{C^{(l)}}\sum_{v=1}^{H^{(l)}D^{(l)}}W_{i,j,u,v}^{(l+1)}\phi\left(A_{j,v}^{(l)}(\boldsymbol{X})\right) \tag{2.48}$$

其中 $i\in\left\{1,\ldots,C^{(l+1)}\right\}$，$u\in\left\{1,\ldots,H^{(l+1)}D^{(l+1)}\right\}$。均值可以计算如下：

$$\mathbb{E}\left[A_{i,u}^{(l+1)}(\boldsymbol{X})\right]=\mathbb{E}\left[b_i^{(l+1)}\right]+\sum_{j=1}^{C^{(l)}}\sum_{v=1}^{H^{(l)}D^{(l)}}\mathbb{E}\left[W_{i,j,u,v}^{(l+1)}\phi\left(A_{j,v}^{(l)}(\boldsymbol{X})\right)\right]=0 \tag{2.49}$$

有协方差：

$$k_u^{(l)}\left(\boldsymbol{X},\boldsymbol{X}'\right)=\mathbb{C}\mathrm{ov}\left[A_{i,u}^{(l)}(\boldsymbol{X}),A_{i,u}^{(l)}(\boldsymbol{X}')\right]=\begin{cases}\sigma_b^2+\dfrac{\sigma_w^2}{C^{(0)}}\displaystyle\sum_{i=1}^{C^{(0)}}\sum_{v\in uth-patch}X_{i,v}X'_{i,v},l=1\\ \sigma_b^2+\sigma_w^2\displaystyle\sum_{v\in uth-patch}V_v^{(l)}\left(\boldsymbol{X},\boldsymbol{X}'\right),l>1\end{cases} \tag{2.50}$$

其中 $V_v^{(l)}\left(\boldsymbol{X},\boldsymbol{X}'\right)=\mathbb{E}\left[\phi\left(A_{j,v}^{(l)}(\boldsymbol{X})\right)\phi\left(A_{j,v}^{(l)}(\boldsymbol{X}')\right)\right]$，与全链接神经网络一样，对于某些激活函数可以给出解析形式。此时通过迭代，即可计算出最后一层的协方差函数 $k_1^{(L+1)}\left(\boldsymbol{X},\boldsymbol{X}'\right)$。

若将 CNN 写成残差形式（Residual CNN），递推形式则变为：

$$\boldsymbol{a}_i^{(l+1)}(\boldsymbol{X}):=\boldsymbol{a}_i^{(l-s)}(\boldsymbol{X})+b_i^{(l+1)}\mathbf{1}+\sum_{j=1}^{C^{(l)}}\boldsymbol{W}_{i,j}^{(l)}\phi\left(\boldsymbol{a}_j^{(l)}(\boldsymbol{X})\right) \tag{2.51}$$

核函数的递归形式则变为：

$$K_u^{(l+1)}\left(\boldsymbol{X},\boldsymbol{X}'\right)=K_u^{(l-s)}\left(\boldsymbol{X},\boldsymbol{X}'\right)+\sigma_b^2+\sigma_w^2\sum_{v\in uth-patch}V_v^{(l)}\left(\boldsymbol{X},\boldsymbol{X}'\right) \tag{2.52}$$

此外，也有考虑包含池化层（Pooling Layer）的 CNN 与 GP 的关系[31]。

深度学习比起传统的机器学习，其强大之处在于对于特征的提取。除了通过神经网络构造核函数，GP 结合神经网络的例子还有深核学习[32]，即在核函数的选择上还是使用常见的核函数，但在计算时，要先将输入 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 进行映射，再计算核函数的值，即：$k(\boldsymbol{x}_i,\boldsymbol{x}_j\,|\,\boldsymbol{\theta})\to k\left(g\left(\boldsymbol{x}_i,\boldsymbol{w}\right),g\left(\boldsymbol{x}_j,\boldsymbol{w}\right)|\,\boldsymbol{\theta},\boldsymbol{w}\right)$，其中 $g(\cdot)$ 为由参数为 $\boldsymbol{w}$ 的神经网络结构生成的非线性映射。为了核函数能够表示更广泛的特征，提高模型的灵活性，Wilson 等（2016）[32]建议使用谱混合核函数（Spectral Mixture Base Kernel）：

$$k_{SM}\left(\boldsymbol{x},\boldsymbol{x}'\,|\,\boldsymbol{\theta}\right)=\sum_{q=1}^{Q}a_q\frac{\left|\Sigma_q\right|^{\frac{1}{2}}}{(2\pi)^{\frac{D}{2}}}\exp\left(-\frac{1}{2}\left\|\Sigma_q^{\frac{1}{2}}\left(\boldsymbol{x}-\boldsymbol{x}'\right)\right\|^2\right)\cos\left\langle\boldsymbol{x}-\boldsymbol{x}',2\pi\boldsymbol{\mu}_q\right\rangle \tag{2.53}$$

其中 $\left\{a_q,\Sigma_q,\boldsymbol{\mu}_q\right\}$ 为核函数的超参数。模型整体为：输入特征经过神经网络提前有效的特征，后经过 GP 输出预测结果。此外该模型训练是端到端的，即同时训练 $\left\{\boldsymbol{\theta},\boldsymbol{w}\right\}$，相关的梯度可由对数边际似然函数（Log Marginal Likelihood）分别求偏

导得[32]。其变种还有变分形式[33]。

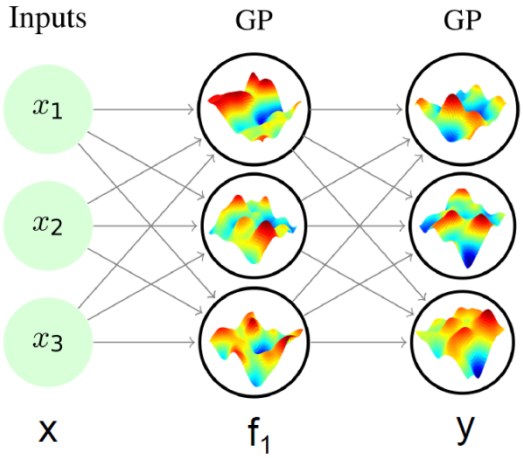

图 2.3 深度高斯过程示意图[149]

由单层无限宽的神经网络是 GP 的观点可以引出另外一个有趣的模型，即深度高斯过程（Deep Gaussian Process）[35],[148]。形象地说，神经网络通过一层一层堆神经元层达到更深的网络结构，深度高斯过程则是一层一层堆叠 GP 走向深度学习的过程，如图 2.3 所示。若使用数学表达式，则 L 层深度高斯过程可以表示为复合多元函数 $f_{1:L}(\boldsymbol{x})=f_L\left(f_{L-1}\left(\cdots f_2\left(f_1(\boldsymbol{x})\right)\cdots\right)\right)$，其基于概率的迭代式也可以用公式表示如下[11][34]：

$$p\left(f_l\mid\boldsymbol{\theta}_l\right)=\mathcal{GP}\left(f_l;0,k_l\right),l=1,\ldots,L \tag{2.54}$$

$$p\left(\boldsymbol{h}_l\mid f_l,\boldsymbol{h}_{l-1},\sigma_l^2\right)=\prod_{i=1}^{n}\mathcal{N}\left(h_{l,i};f_l\left(h_{l-1,i}\right),\sigma_l^2\right),\quad \boldsymbol{h}_{1,i}=\boldsymbol{x}_i \tag{2.55}$$

$$p\left(\boldsymbol{y}\mid f_L,\boldsymbol{h}_{L-1},\sigma_L^2\right)=\prod_{i=1}^{n}\mathcal{N}\left(y_i;f_L\left(\boldsymbol{h}_{L-1,i}\right),\sigma_L^2\right) \tag{2.56}$$

其中 $\boldsymbol{h}_l$ 为第 l 层隐藏层的输出。式（2.54）可以看作是隐藏层的先验，式（2.55）则为后验。

因为中间层的输入都是随机变量，所以输出的后验分布不总是边际正态的。因此，有很多推断的方法被应用于 DGP 当中，得益于近似推断，稀疏近似的方法也可被使用，即使用有限 M 个函数值的诱导输出（Inducing Outputs）$u(\boldsymbol{z})$ 来表示无限维的向量 $f(\boldsymbol{x})$。推断过程中，边际似然与预测的后验分布不可避免地需要将 $\boldsymbol{u}$ 作为被积函数积分掉，现已有使用变分推断[35]、期望传播[34]的近似推断。以期望传播算法为例，考虑双层的 DGP，需要计算积分：

$$\begin{aligned}Z&=\int p\left(y\mid\boldsymbol{x},\boldsymbol{u}\right)q^{\backslash 1}\left(\boldsymbol{u}\right)d\boldsymbol{u}\\&=\int p\left(y\mid h_1,\boldsymbol{u}_2\right)q^{\backslash 1}\left(\boldsymbol{u}_2\right)dh_1d\boldsymbol{u}_2\int p\left(h_1\mid\boldsymbol{x},\boldsymbol{u}_1\right)q^{\backslash 1}\left(\boldsymbol{u}_1\right)d\boldsymbol{u}_1\end{aligned} \tag{2.57}$$

其中 $q^{\backslash 1}\left(\boldsymbol{u}\right)=q^{\backslash i}\left(\boldsymbol{u}\right)\propto q\left(\boldsymbol{u}\right)/\tilde{t}_i\left(\boldsymbol{u}\right)$ 因为对称性，$\tilde{t}_i\left(\boldsymbol{u}\right)$ 为分布 $q\left(\boldsymbol{u}\right)\propto p\left(\boldsymbol{u}\right)\prod_i\tilde{t}_i\left(\boldsymbol{u}\right)$

[11]该式考虑的为监督学习，经典文献[35]中考虑的是非监督学习，即输入为隐变量。

的第 i 个因子。首先对诱导点求积分，得 $Z=\int q(y|h_1)q(h_1)dh_1$，其中 $q(h_1)=\mathcal{N}(h_1;m_1,v_1)$，$q(y|h_1)=\mathcal{N}(y|h_1;m_{2|h_1},v_{2|h_1})$[12]。使用叠期望定律与叠方差定理（Law of Iterated Expectation and Variance），有

$$m_2=\mathbb{E}_{q(h_1)}\left[m_{2|h_1}\right] \tag{2.58}$$

$$v_2=\mathbb{E}_{q(h_1)}\left[v_{2|h_1}\right]+\mathbb{V}\mathrm{ar}_{q(h_1)}\left[m_{2|h_1}\right] \tag{2.59}$$

期望需要核矩阵在输入的高斯分布上求积分，对于很大一部分核函数而言，该积分是可解析的[34]。而对于预测分布 $p(y_*|x_*,\boldsymbol{X},\boldsymbol{Y})\approx\int p(y_*|x_*,\boldsymbol{u})q(\boldsymbol{u}|\boldsymbol{X},\boldsymbol{Y})d\boldsymbol{u}$，同样可以用上述方法，使用高斯分布近似。

考虑深度高斯过程的宽度与深度，其分别对应深度神经网络的宽度与深度。如前面所提及的深度无限宽神经网络一样，深度高斯过程在宽度趋于无穷的时候，会收敛到一个单层的高斯过程[37]。并且，Pleiss 和 Cunningham（2021）[37]还描述了深度与宽度的“对抗性”效果：宽度提升了模型的高斯性，而深度则强调的是非高斯性（如：尖峰、厚尾、偏态）。然而当深度逐渐加深时，模型会渐渐丧失自由度，当层数趋于无穷时，仅有一个自由度起作用[38]。例如输入有 10 个维度时，当层数趋于无穷时只有其中一个维度在起作用。一个中间层连接输入的结构可以缓解该问题，其结构如下所示：

$$f_{1:L}(\boldsymbol{x})=f_L\left(f_{1:L-1}(\boldsymbol{x}),\boldsymbol{x}\right),\forall L \tag{2.60}$$

此外，Dunlop 等（2018）[39]为多层的高斯过程提供了一个更一般的、具有马尔科夫结构的框架，并通过遍历性（Ergodicity）分析了深度高斯过程的有效深度。

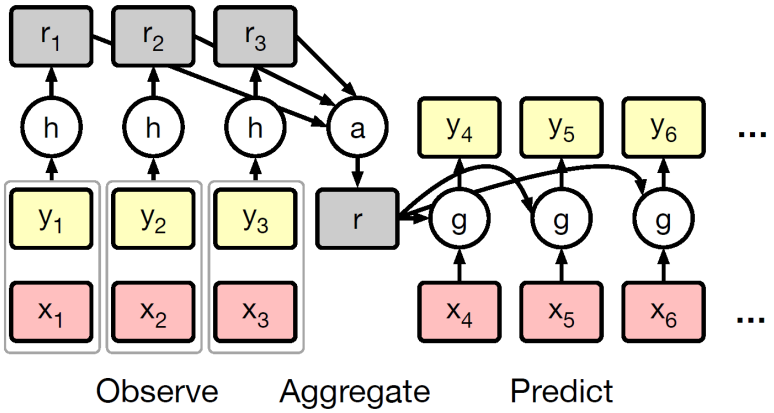

图 2.4　条件神经过程模型图例[40]，h 和 g 分别为编码器和解码器。

另一种与神经网络结合的方法为神经过程（Neural Process）[40]，将元学习（Meta-Learning）的思想应用于监督学习[13]。特别地，当 GPR 使用 GP 的先验知识时，神经过程 Q_θ 则使用参数 θ 参数化随机过程 Q，其中的 θ 类比于元学习中用

[12] 分布的均值与方差的具体结果在此不给出，可参考文献[34]。
[13] 本文仅介绍条件神经过程[40]。

来生成算法的参数。模型的结构如图 2.4 所示，具体的数学公式可表示如下：

$$r_i = h_\theta\left(\boldsymbol{x}_i, \boldsymbol{y}_i\right), \forall\left(\boldsymbol{x}_i, \boldsymbol{y}_i\right) \in O \tag{2.61}$$

$$r = r_1 \oplus r_2 \oplus \cdots \oplus r_n \tag{2.62}$$

$$\boldsymbol{\phi}_i = g_\theta\left(\boldsymbol{x}_i, r\right), \forall\left(\boldsymbol{x}_i\right) \in T \tag{2.63}$$

其中，编码器 $h_\theta : X \times Y \to \mathbb{R}^{in}$ 和解码器 $g_\theta : X \times \mathbb{R}^{in} \to \mathbb{R}^{de}$ 为神经网络，维数 in 和 de 一般需要设置，X 和 Y 在这里分别对应输入输出域；$\oplus$ 为可交换的算子，将多个 $\mathbb{R}^{in}$ 上的向量映射到一个 $\mathbb{R}^{in}$ 上的向量上，一般可令 $r = \left(\sum_n r_i\right)/n$；$\boldsymbol{\phi}_i$ 则为 $Q_\theta\left(f\left(x_i\right) | O, x_i\right) = Q\left(f\left(x_i\right) | \boldsymbol{\phi}_i\right)$ 的参数，对于回归任务，使用 $\boldsymbol{\phi}_i = \{\mu_i, \sigma_i^2\}$ 来参数化分布 $\mathcal{N}\left(\mu_i, \sigma_i^2\right)$ 的均值与方差；在此用 O 和 T 分别表示训练集与测试集。

该模型训练也如元学习一样，从训练集 O 中取出一部分 O_N，训练 Q_θ 以预测 O。一般地，令 $f \sim P$，$O = \left\{\left(x_i, y_i\right)\right\}_{i=0}^{n-1}$，$N \sim \text{uniform}[0, \ldots, n-1]$ 为离散均匀分布，此时可得训练时用于训练的样本 $O_N = \left\{\left(x_i, y_i\right)\right\}_{i=0}^{N} \in O$。最大化对数概率分布

$$L(\theta) = E_{f \sim P}\left[E_N\left[\log Q_\theta\left(\left\{y_i\right\}_{i=0}^{n-1} | O_N, \left\{x_i\right\}_{i=0}^{n-1}\right)\right]\right] \tag{2.64}$$

以训练参数。特别地，对于回归任务而言，上式可以理解为：由 O_N 生成每个 x_i 对应的分布参数 $\boldsymbol{\phi}_i$，并由该分布与已知的 y_i 计算类似于似然函数的指标。

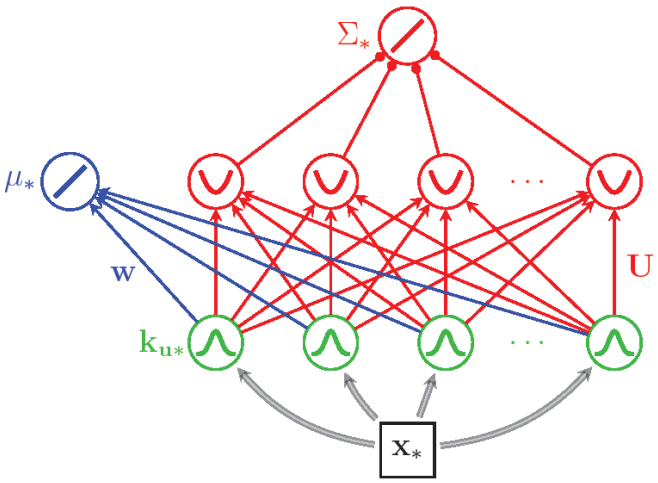

图 2.5　稀疏高斯过程回归的神经网络形式[43]

此外，（稀疏）高斯过程回归也可以看做有限宽的神经网络[43]。考虑诱导点 $u(z)$ 及其位置 z，有 $\left[\boldsymbol{K}_{fu}\right]_{ij} = k\left(\boldsymbol{x}_i, \boldsymbol{z}_j\right)$ 和 $\left[\boldsymbol{K}_{uu}\right]_{ij} = k\left(\boldsymbol{z}_i, \boldsymbol{z}_j\right)$。在预测点 $\boldsymbol{x}_*$ 上有预测分布 $q\left(y_*\right) = \mathcal{N}\left(y_*; \mu_*, \Sigma_*\right)$，其中

$$\mu_* = \boldsymbol{k}_{u*}^\top\left(\boldsymbol{K}_{uf}\boldsymbol{K}_{fu} + \sigma_\varepsilon^2\boldsymbol{K}_{uu}\right)^{-1}\boldsymbol{K}_{uf}\boldsymbol{y} \tag{2.65}$$

$$\Sigma_* = k_{**} - \boldsymbol{k}_{u*}^\top\boldsymbol{K}_{uu}^{-1}\boldsymbol{k}_{u*} + \boldsymbol{k}_{u*}^\top\left(\sigma_\varepsilon^{-2}\boldsymbol{K}_{uf}\boldsymbol{K}_{fu} + \boldsymbol{K}_{uu}\right)^{-1}\boldsymbol{k}_{u*} + \sigma_\varepsilon^2 \tag{2.66}$$

将上式的均值改写为 $\mu_* = \sum_j w_j k\left(\boldsymbol{z}_j, \boldsymbol{x}_*\right)$，其中 $\boldsymbol{w} = \left(\boldsymbol{K}_{uf}\boldsymbol{K}_{fu} + \sigma_\varepsilon^2 \boldsymbol{K}_{uu}\right)^{-1}\boldsymbol{K}_{uf}\boldsymbol{y}$。给定激活函数 $\phi_j\left(\boldsymbol{x}_*\right) = k\left(\boldsymbol{z}_j, \boldsymbol{x}_*\right)$，均值则可看成为一个简单的神经网络：输入 $\boldsymbol{x}_*$ 首先经过多个激活函数生成多个神经元，然后通过线性组合生成一个输出均值的神经元。

同样对于方差可以改写为 $\Sigma_* = k_{**} - \boldsymbol{k}_{u*}^\top \boldsymbol{A}\boldsymbol{k}_{u*} + \sigma_\varepsilon^2$，其中正定矩阵 $\boldsymbol{A}$ 可分解为 $\boldsymbol{A} = \boldsymbol{U}^\top\boldsymbol{U}$，则 $\boldsymbol{k}_{u*}^\top\boldsymbol{A}\boldsymbol{k}_{u*} = \left(\boldsymbol{U}\boldsymbol{k}_{u*}\right)^\top\left(\boldsymbol{U}\boldsymbol{k}_{u*}\right) = \sum_j\left(\boldsymbol{U}\boldsymbol{k}_{u*}\right)^2 = \sum_j \psi_j$，其中 $\psi_j = \left(\boldsymbol{U}\boldsymbol{\phi}\right)_j^2$。由此可见，方差可以表示为第一层为 $\boldsymbol{\phi}$，第二层为 $\boldsymbol{\Psi}$ 的双层神经网络结构，神经元个数则等于诱导点 u 的个数。

2.4 与其他模型、算法的联系

2.4.1 贝叶斯优化

高斯过程回归的一个经典应用是贝叶斯优化（Bayesian Optimization），将超参数作为 $\boldsymbol{x}$ 输入，输出 f 则为模型待优化的目标函数。给定训练的数据集 $\mathcal{D}_{t-1}\left\{\left(\boldsymbol{x}_1, y_1\right),\left(\boldsymbol{x}_2, y_2\right),\ldots,\left(\boldsymbol{x}_{t-1}, y_{t-1}\right)\right\}$，回归的任务为预测 $\boldsymbol{x}_t$ 点上的值 f_t，而贝叶斯优化则关注于已知数据集时搜寻一个可能使目标函数更优的值，即：$\boldsymbol{x}_t = \arg\min \alpha_t\left(\boldsymbol{x}\right)$，其中 $\alpha_t\left(\boldsymbol{x}\right)$ 为搜寻的标准。因为给定数据集时，目标函数是服从高斯过程的，即包含均值与不确定性的方差，所以不同的搜寻标准会得到不一样的迭代超参数，如：从后验分布中抽样出一个样本轨道 $\alpha_t(\boldsymbol{x}) = \hat{f}_t$。显然，不需要求梯度的优化提供了许多便利，但优化器受限于高斯过程回归本身。

2.4.2 线性贝叶斯

线性贝叶斯模型在 GPML[9]中被称为是以权重视角理解高斯过程回归。考虑带映射的线性贝叶模型

$$f\left(\boldsymbol{x}\right) = \phi\left(\boldsymbol{x}\right)^\top \boldsymbol{w}, y = f\left(\boldsymbol{x}\right) + \varepsilon \tag{2.67}$$

其中，$\phi\left(\boldsymbol{x}\right)$ 为从 D 维输入向量到 N 维特征空间的映射。假定权重 $\boldsymbol{w} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{\Sigma}_p\right)$，并且有 $p\left(\boldsymbol{y}\,|\,\boldsymbol{X}, \boldsymbol{w}\right) = \mathcal{N}\left(\boldsymbol{\Phi}^\top\boldsymbol{w}, \sigma_\varepsilon^2\boldsymbol{I}\right)$，其中 $\boldsymbol{\Phi} = \phi\left(\boldsymbol{X}\right)$，可得权重的后验分布

$$p\left(\boldsymbol{w}\,|\,\boldsymbol{y}, \boldsymbol{\Phi}\right) = \frac{p\left(\boldsymbol{y}\,|\,\boldsymbol{w}, \boldsymbol{\Phi}\right)p\left(\boldsymbol{w}\right)}{p\left(\boldsymbol{y}\,|\,\boldsymbol{\Phi}\right)} = \mathcal{N}\left(\frac{1}{\sigma_\varepsilon^2}\boldsymbol{A}^{-1}\boldsymbol{\Phi}\boldsymbol{y}, \boldsymbol{A}^{-1}\right) \tag{2.68}$$

其中 $p\left(\boldsymbol{y}\,|\,\boldsymbol{\Phi}\right)$ 为标准化常数，$\boldsymbol{A} = \sigma_\varepsilon^{-2}\boldsymbol{\Phi}\boldsymbol{\Phi}^\top + \boldsymbol{\Sigma}_p^{-1}$。给定预测点 $\boldsymbol{x}_*$，可得预测分布

$$\begin{aligned} p\left(f_*\,|\,\boldsymbol{x}_*, \boldsymbol{\Phi}, \boldsymbol{y}\right) &= \int p\left(f_*\,|\,\boldsymbol{x}_*, \boldsymbol{w}\right)p\left(\boldsymbol{w}_*\,|\,\boldsymbol{\Phi}, \boldsymbol{y}\right)d\boldsymbol{w} \\ &= \mathcal{N}\left(\frac{1}{\sigma_\varepsilon^2}\phi\left(\boldsymbol{x}_*\right)^\top\boldsymbol{A}^{-1}\boldsymbol{X}\boldsymbol{y}, \phi\left(\boldsymbol{x}_*\right)^\top\boldsymbol{A}^{-1}\phi\left(\boldsymbol{x}_*\right)\right) \end{aligned} \tag{2.69}$$

定义核函数 $k\left(\boldsymbol{x}, \boldsymbol{x}'\right) = \phi\left(\boldsymbol{x}\right)^\top\boldsymbol{\Sigma}_p\phi\left(\boldsymbol{x}'\right)$，有 $\boldsymbol{K} = \boldsymbol{\Phi}^\top\boldsymbol{\Sigma}_p\boldsymbol{\Phi}$，可以看出预测结果与高斯

过程回归相同。

2.4.3 核岭回归

考虑一个特殊的带正则项的泛函[9]：

$$J[f]=\frac{1}{2}\|f\|_H^2+\frac{1}{2\sigma_\varepsilon^2}\sum_{i=1}^{n}\left(y_i-f\left(\boldsymbol{x}_i\right)\right)^2 \tag{2.70}$$

根据表示理论（Representer Theorem），可以得到最优的预测函数形式为 $f\left(\boldsymbol{x}\right)=\sum_{i=1}^{n}\alpha_i k\left(\boldsymbol{x},\boldsymbol{x}_i\right)$，并且令 $\left\langle k\left(\cdot,\boldsymbol{x}_i\right),k\left(\cdot,\boldsymbol{x}_j\right)\right\rangle_H=k\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right)$，有

$$\begin{aligned}J[\boldsymbol{\alpha}]&=\frac{1}{2}\boldsymbol{\alpha}^\top\boldsymbol{K}\boldsymbol{\alpha}+\frac{1}{2\sigma_\varepsilon^2}\left|\boldsymbol{y}-\boldsymbol{K}\boldsymbol{\alpha}\right|^2\\&=\frac{1}{2}\boldsymbol{\alpha}^\top\left(\boldsymbol{K}+\frac{1}{\sigma_\varepsilon^2}\boldsymbol{K}^2\right)\boldsymbol{\alpha}-\frac{1}{\sigma_\varepsilon^2}\boldsymbol{y}^\top\boldsymbol{K}\boldsymbol{\alpha}+\frac{1}{2\sigma_\varepsilon^2}\boldsymbol{y}^\top\boldsymbol{y}\end{aligned} \tag{2.71}$$

最小化 J 可得最优权重 $\hat{\alpha}=\left(\boldsymbol{K}+\sigma_\varepsilon^2\boldsymbol{I}\right)^{-1}\boldsymbol{y}$，即在预测点 $\boldsymbol{x}_*$ 上有预测 $\hat{f}\left(\boldsymbol{x}_*\right)=k\left(\boldsymbol{X},\boldsymbol{x}_*\right)^\top\left(\boldsymbol{K}+\sigma_\varepsilon^2\boldsymbol{I}\right)^{-1}\boldsymbol{y}$，其与高斯过程回归的预测均值相同。

2.4.4 支持向量回归

对于线性回归任务，支持向量回归的优化目标为[9]：

$$\frac{1}{2}|\boldsymbol{w}|^2+C\sum_{i=1}^{n}g_\varepsilon\left(y_i-f_i\right) \tag{2.72}$$

其中权重 $\boldsymbol{w}$ 有高斯先验，参数 $C>0$，$\varepsilon-$不敏感的损失函数为

$$g_\varepsilon\left(z\right)=\begin{cases}|z|-\varepsilon,|z|\geq\varepsilon\\0,|z|<\varepsilon\end{cases} \tag{2.73}$$

给定预测点 $\boldsymbol{x}_*$ 时，由二次规划问题可得预测值 $f\left(\boldsymbol{x}_*\right)=\sum_{i=1}^{n}\alpha_i\boldsymbol{x}_i\boldsymbol{x}_*$，特别地，使用核函数时有预测 $f\left(\boldsymbol{x}_*\right)=\sum_{i=1}^{n}\alpha_i k\left(\boldsymbol{x}_i,\boldsymbol{x}_*\right)$，与高斯过程回归均值预测相似。

2.4.5 样条

样条法最初被用于一维的插值与平滑的任务中，目标是找到一个二阶导有定义的 Sobolev 函数空间上的函数 f 使得

$$\int_a^b\left(f^{(m)}\left(x\right)\right)^2dx \tag{2.74}$$

最小化[44]。当 $m=2$ 时，存在唯一一个显式的、有限维的最小值点，有

$$\begin{aligned}&f\left(x\right)=\sum_{j=0}^{1}\beta_j x^j+\sum_{i=1}^{n}\alpha_i\left(x-x_i\right)_+^3,\\&\left(z\right)_+=\begin{cases}z,z>0\\0,z\leq0\end{cases}\end{aligned} \tag{2.75}$$

是一个结点在不重复的 x_i 上的自然三次样条。具体地，考虑一个单变量样条平

滑问题，其优化的泛函为：

$$\sum_{i=1}^{n}\left(f\left(x_i\right)-y_i\right)^2+\lambda\int_2^1\left(f''\left(x\right)\right)^2dx \tag{2.76}$$

其中λ为给定的光滑参数，$\lambda=0$时为插值，$\lambda=\infty$时为简单的最小二乘拟合。其解形式同式（2.75）。

考虑一个生成模型：

$$g\left(x\right)=\sum_{j=0}^{1}\beta_j x^j+f\left(x\right) \tag{2.77}$$

其中$\beta\sim\mathcal{N}\left(0,\sigma_\beta^2 I\right)$，$f\left(x\right)$为零均值的高斯过程，其协方差为$\sigma_f^2 k_{sp}\left(x,x'\right)$，其中

$$k_{sp}\left(x,x'\right)\triangleq\int_0^1\left(x-u\right)_+\left(x'-u\right)_+du=\frac{\left|x-x'\right|v^2}{2}+\frac{v^3}{3} \tag{2.78}$$

$v=\min\left(x,x'\right)$。由高斯过程回归给出预测均值

$$\overline{f}\left(x_*\right)=k\left(\boldsymbol{X},x_*\right)^\top\boldsymbol{K}_y^{-1}\left(\boldsymbol{y}-\boldsymbol{H}^\top\overline{\boldsymbol{\beta}}\right)+\boldsymbol{h}\left(x_*\right)^\top\overline{\boldsymbol{\beta}} \tag{2.79}$$

其中$\boldsymbol{K}_y=\sigma_f^2 k_{sp}\left(\boldsymbol{X},\boldsymbol{X}\right)+\sigma_\varepsilon^2\boldsymbol{I}$为协方差矩阵，$\boldsymbol{h}\left(x\right)=\left(1,x\right)^\top$且矩阵$\boldsymbol{H}$的第$i$个列向量为$\boldsymbol{h}\left(x_i\right)$，$\overline{\boldsymbol{\beta}}=\left(\boldsymbol{H}\boldsymbol{K}_y^{-1}\boldsymbol{H}^\top\right)^{-1}\boldsymbol{H}\boldsymbol{K}_y^{-1}\boldsymbol{y}$。可以看出，预测均值的第一部分对应式（2.75）的三次断点部分，因为$k\left(x_*\right)$为分段三次多项式，而第二部分$\boldsymbol{h}\left(x_*\right)^\top\overline{\boldsymbol{\beta}}$则对应了线性部分。此外，Raket（2021）[45]考虑了高斯过程先验与平滑样条之间的联系，也考虑了高斯过程与微分方程之间的联系。

2.4.6 微分方程

一个高斯过程与微分方程结合的模型名为隐力模型（Latent Force Models）[46]，它借助了高斯过程来描述力平衡的微分方程，其中的核函数由微分方程导出。传统的机械论模型精确地考虑了所有力的作用，而隐力模型借助于核函数则简化地考虑了 Q 个简化的、等价的隐力来描述物理系统。给定输入t与 D 维输出[14]$\left\{f_d\left(t\right)\right\}_{d=1}^{D}$，隐力模型给出输出$f_d\left(t\right)=\sum_{q=1}^{Q}S_{d,q}\int_0^t G_d\left(t-\tau\right)u_q\left(\tau\right)d\tau$，其中$G_d\left(\cdot\right)$为对应于确定的线性微分方程的格林函数，$u_q\left(t\right)\sim GP\left(0,k_q\left(t,t'\right)\right)$为第 q 个隐力对应的高斯过程先验，$S_{d,q}$则为第 q 个隐力对第 d 维输出贡献的权重。此时，输出间的协方差可以广义地表示为：

$$k_{f_d,f_{d'}}\left(t,t'\right)=\sum_{q=1}^{Q}S_{d,q}S_{d',q}\int_0^t G_d\left(t-\tau\right)\int_0^{t'}G_{d'}\left(t'-\tau'\right)k_q\left(\tau,\tau'\right)d\tau'd\tau \tag{2.80}$$

2.4.7 马尔科夫过程

高斯马尔科夫过程[48]是一个高斯过程f，但概率$p\left(f\left(\boldsymbol{x}_i\right)|f\left(\boldsymbol{x}_{n/i}\right)\right)$只取决

[14] 可看作多目标回归。

于 $\boldsymbol{x}_i$ 的邻居（Neighbours）集合 $\partial \boldsymbol{x}_i$ 。显然，使用高斯-马尔科夫过程可以通过稀疏矩阵的形似减少计算量。特别地，使用特殊的马顿核的高斯过程是线性泛函随机偏微分方程（SPDE）的解，其中方程为

$$\left(\kappa^2-\Delta\right)^{\alpha/2} f\left(u\right)=W\left(u\right),u\in R^d,\alpha=v+d/2,\kappa>0,v>0 \quad (2.81)$$

其中更新过程 W 是单位方差的空间高斯白噪声，$\Delta=\sum_{i=1}^{d}\partial^2/\partial x_i^2$ 是拉普拉斯算子。

此外，一些高斯过程模型与其他模型之间的比较可见 Rasmussen（1999）[49]。

3 大数据高斯过程回归综述

标准的 GP 在处理小样本任务时表现得非常好，但难以处理数据量大于 $\mathcal{O}(10^4)$ 的数据集。因为训练模型时，计算 $n\times n$ 维矩阵的逆及其行列式都会产生 $\mathcal{O}(n^3)$ 的时间复杂度，并且在预测时，矩阵向量化的操作也花费了 $\mathcal{O}(n^2)$，同时计算所需的内存大小也需要 $\mathcal{O}(n^2)$[14]。在大数据时代，海量的数据通过数据量增强了模型的预测能力，但同时也要求模型拥有在有限的时间内处理大量数据的能力。标准的 GP 处理大数据能力较差，因此如何改进 GP 使其能够处理大数据又不失预测精度则成了当前主要要解决的问题（后将高斯过程回归模型在大数据集上的拓展方法简称为“加速方法”）。

为了解决大数据集上 GP 的使用问题，近二十年已经提出了许多有效的加速方法，也已有多篇综述来总结这些方法，如 Liu 等（2019，2020）[50][74]。但由于作为综述类文章发表，需要一个统一的框架来归纳已有的模型，并且一些新颖的文章中的相关工作（Related Work）部分，也都只考虑了部分加速方法，所以在本部分内容当中，损失了一些对文章理解的深度以及优美的统一框架，而追求加速方法的广度，更有利于读者了解高斯过程回归模型在大数据集上的拓展加速方法。因为该领域的研究依旧非常活跃，且尚未有一个方法能够满足工业应用的需求，故本文对此方面的总结可能存在遗漏，目的为将读者过度到感兴趣的领域从而进行深入研究[15]。本文具体的综述框架如图 3.1 所示：

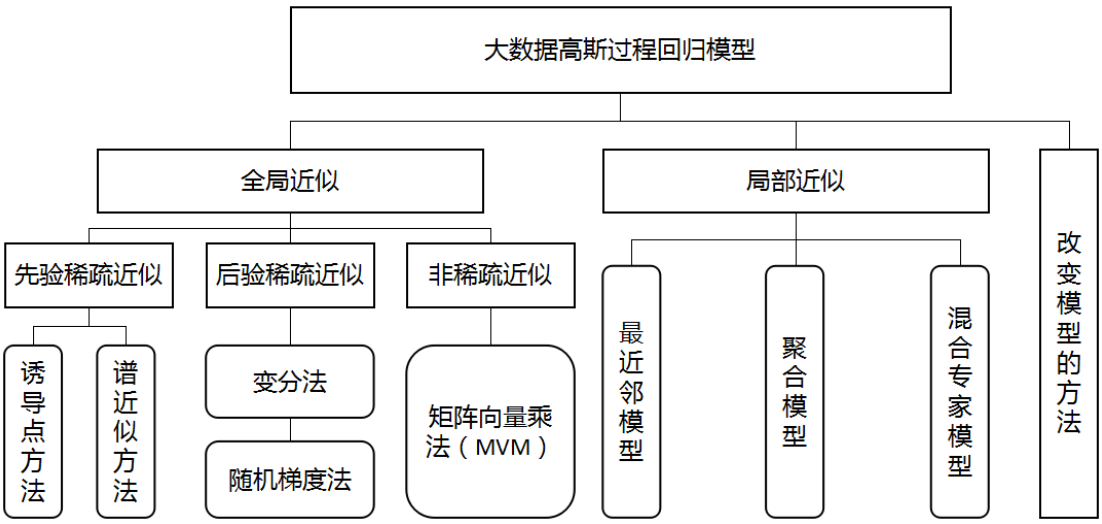

图 3.1 大数据高斯过程回归模型的拓展方法框架

[15]如:高斯过程回归属于核方法，但加速方法本文只参考了应用于高斯过程模型的拓展方法，对于那些有效的、但是尚未与高斯过程模型结合的方法本文尚未提及。

参考 Liu 等（2019）[50]，本文将加速方法的主体同样主要分为全局近似与局部近似两个部分，并在此基础上进行了扩充。其中，简单地举个例子，全局近似为不改变模型推断框架时的加速方法，如选取一个小的子集来总结整个训练集并用以训练；局部近似则使用了“分而治之”的思想，将整个数据集划分为 M 个子集，最后的预测是通过整合 M 个子模型的预测而得的。此外，我们考虑了模型的训练方法、其他方法以及近似模型的收敛率相关内容。但由于后期的文章通常会结合多种加速方法，故本文对于文章的分类并未达到非常高的精度。

3.1 全局近似

本部分内容主要包括稀疏近似方法和非稀疏近似方法两块部分，其中稀疏近似方法又主要包括近似先验与近似推理的两部分。而非稀疏近似则主要包括线性系统迭代法以及其他的矩阵分解方法。

3.1.1 先验稀疏近似

稀疏近似方法主要是近似核矩阵，使得其求逆与求行列式的时间消耗降低。对于“近似先验，精确后验”的部分，主要有“诱导点”方法（inducing points，可以理解为伪输入、支持输入）、谱近似方法（以核矩阵为目标）。对于“精确先验，近似后验”的部分，文章主要考虑变分方法。

3.1.1.1 诱导点方法

首先考虑“诱导点”方法。将诱导点 $\boldsymbol{f}_m$ 对应的输入记为 $\boldsymbol{X}_m$，给定独立性假设 $\boldsymbol{f} \perp f_* \mid \boldsymbol{f}_m$，联合先验 $p(\boldsymbol{f}, f_*)$ 可以通过边缘化隐变量 $\boldsymbol{f}_m$ 得到

$$p(\boldsymbol{f}, f_*) = \int p(\boldsymbol{f} \mid \boldsymbol{f}_m) p(f_* \mid \boldsymbol{f}_m)\, p(\boldsymbol{f}_m)\, d\boldsymbol{f}_m \tag{3.1}$$

给定 Nystrom 记号 $\boldsymbol{Q}_{ab} = \boldsymbol{K}_{am}\boldsymbol{K}_{mm}^{-1}\boldsymbol{K}_{mb}$ [52]，上式中的训练条件概率与测试条件概率可以分别表示为

$$p(\boldsymbol{f} \mid \boldsymbol{f}_m) = \mathcal{N}\left(\boldsymbol{f} \mid \boldsymbol{K}_{nm}\boldsymbol{K}_{mm}^{-1}\boldsymbol{f}_m, \boldsymbol{K}_{nn} - \boldsymbol{Q}_{nn}\right) \tag{3.2}$$

$$p(f_* \mid \boldsymbol{f}_m) = \mathcal{N}\left(f_* \mid \boldsymbol{k}_{*m}\boldsymbol{K}_{mm}^{-1}\boldsymbol{f}_m, k_{**} - Q_{**}\right) \tag{3.3}$$

该类方法之所以被称为“诱导点”方法，是因为 $\boldsymbol{f}$ 和 f_* 只能通过诱导点 $\boldsymbol{f}_m$ 来交流信息，可以由独立性假设看出，因此称 $\boldsymbol{f}_m$ “诱导”出了训练条件分布与测试条件分布的相关性。

为了降低运算的时间复杂度，先验可以有如下近似

$$p(\boldsymbol{f}, f_*) \approx q(\boldsymbol{f}, f_*) = \int q(\boldsymbol{f} \mid \boldsymbol{f}_m) q(f_* \mid \boldsymbol{f}_m)\, p(\boldsymbol{f}_m)\, d\boldsymbol{f}_m \tag{3.4}$$

由此，为了避免全样本矩阵 $\boldsymbol{K}_{nn}$ 和 K_{**} 的计算，条件概率可以分别近似如下

$$q(\boldsymbol{f}\mid\boldsymbol{f}_m)=\mathcal{N}\left(\boldsymbol{f}\mid\boldsymbol{K}_{nm}\boldsymbol{K}_{mm}^{-1}\boldsymbol{f}_m,\tilde{\boldsymbol{Q}}_{nn}\right) \tag{3.5}$$

$$q(f_*\mid\boldsymbol{f}_m)=\mathcal{N}\left(f_*\mid\boldsymbol{k}_{*m}\boldsymbol{K}_{mm}^{-1}\boldsymbol{f}_m,\tilde{Q}_{**}\right) \tag{3.6}$$

同样边缘化 $\boldsymbol{f}_m$，用于训练的对数边际似然 $\log p(\boldsymbol{y})$ 可被近似为

$$\log q(\boldsymbol{y})=-\frac{1}{2}\log(2\pi)-\frac{1}{2}\log\left|\tilde{\boldsymbol{Q}}_{nn}+\boldsymbol{Q}_{nn}+\sigma_\varepsilon^2\boldsymbol{I}\right|-\frac{1}{2}\boldsymbol{y}^\top\left(\tilde{\boldsymbol{Q}}_{nn}+\boldsymbol{Q}_{nn}+\sigma_\varepsilon^2\boldsymbol{I}\right)^{-1}\boldsymbol{y} \tag{3.7}$$

此类方法虽然有着不同的出发点，但大多可被纳入该框架下，具体表现为给定不同假设以及选用不同的 $\tilde{\boldsymbol{Q}}_{nn}$ 和 $\tilde{Q}_{**}$，使得对 $\boldsymbol{K}_{nn}$ 的处理转为对 $\boldsymbol{K}_{mm}$ 的处理。

经典的诱导点方法[16]有 SoR（Subset of Regressors）、DTC（Deterministic Training Conditional）、FI(T)C（Fully Independent (Training) Conditional）、PI(T)C（Partially Independent (Training) Conditional），对此，本文只给出不同模型的 $\tilde{\boldsymbol{Q}}_{nn}$ 和 $\tilde{Q}_{**}$，细节可参考如 Quinonero-Candela 和 Rasmussen（2005）[51]。其中 SoR 等价于对训练数据与测试数据使用 Nystrom 近似，有 $\tilde{\boldsymbol{Q}}_{nn}=\boldsymbol{0}$ 和 $\tilde{Q}_{**}=0$，以概率表示如下

$$q_{SoR}(\boldsymbol{f}\mid\boldsymbol{f}_m)=\mathcal{N}\left(\boldsymbol{f}\mid\boldsymbol{K}_{nm}\boldsymbol{K}_{mm}^{-1}\boldsymbol{f}_m,\boldsymbol{0}\right) \tag{3.8}$$

$$q_{SoR}(f_*\mid\boldsymbol{f}_m)=\mathcal{N}\left(f_*\mid\boldsymbol{k}_{*m}\boldsymbol{K}_{mm}^{-1}\boldsymbol{f}_m,0\right) \tag{3.9}$$

并且，SoR 模型的训练条件概率与测试条件概率中的核函数是一致的，等价于退化的 GP，其核函数至多为 m 阶有 $k(\boldsymbol{x}_i,\boldsymbol{x}_j)=k(\boldsymbol{x}_i,\boldsymbol{X}_m)\boldsymbol{K}_{mm}^{-1}k(\boldsymbol{X}_m,\boldsymbol{x}_j)$。明显地，与标准的 GP 相比，SoR 受限于 m 个自由度，其预测方差通常会偏低。此时，将近似的测试条件概率还原为精确的概率时，可以得到 DTC 的条件概率

$$q_{DTC}(\boldsymbol{f}\mid\boldsymbol{f}_m)=\mathcal{N}\left(\boldsymbol{f}\mid\boldsymbol{K}_{nm}\boldsymbol{K}_{mm}^{-1}\boldsymbol{f}_m,\boldsymbol{0}\right) \tag{3.10}$$

$$q_{DTC}(f_*\mid\boldsymbol{f}_m)=p(f_*\mid\boldsymbol{f}_m) \tag{3.11}$$

此方法的改进方向基于隐变量的确定性映射 $\boldsymbol{f}=\boldsymbol{K}_{nm}\boldsymbol{K}_{mm}^{-1}\boldsymbol{f}_m$，若要进一步考虑 $\boldsymbol{f}$ 的方差时，假设不考虑 $\boldsymbol{f}_i$ 之间的相关性，则可以得到 FITC 的训练条件概率

$$q_{FITC}(\boldsymbol{f}\mid\boldsymbol{f}_m)=\mathcal{N}\left(\boldsymbol{f}\mid\boldsymbol{K}_{nm}\boldsymbol{K}_{mm}^{-1}\boldsymbol{f}_m,\operatorname{diag}\left[\boldsymbol{K}_{nn}-\boldsymbol{Q}_{nn}\right]\right) \tag{3.12}$$

此时若同样对测试条件概率，可以得到 FIC 模型

$$q_{FIC}(\boldsymbol{f}_*\mid\boldsymbol{f}_m)=\mathcal{N}\left(f_*\mid\boldsymbol{k}_{*m}\boldsymbol{K}_{mm}^{-1}\boldsymbol{f}_m,\operatorname{diag}\left[K_{**}-Q_{**}\right]\right) \tag{3.13}$$

之后，还可以从方差入手，进一步提升保留的信息，得到相应的 PITC 和 PIC 模型[53]。先将 $\boldsymbol{f}$ 或 $\boldsymbol{f}_*$ 进行分区，保留区块内的协方差而忽略区与区之间的相关信息，有条件概率

[16]BCM 也可被纳入该框架中，只需令诱导点 $\boldsymbol{f}_m=\boldsymbol{f}_*$，但本文将 BCM 纳入了局部近似的类别中。

$$q_{PITC}\left(\boldsymbol{f}\mid\boldsymbol{f}_m\right)=\mathcal{N}\left(\boldsymbol{f}\mid\boldsymbol{K}_{nm}\boldsymbol{K}_{mm}^{-1}\boldsymbol{f}_m,\operatorname{blockdiag}\left[\boldsymbol{K}_{nn}-\boldsymbol{Q}_{nn}\right]\right) \quad (3.14)$$

$$q_{PITC}\left(\boldsymbol{f}_*\mid\boldsymbol{f}_m\right)=p\left(\boldsymbol{f}_*\mid\boldsymbol{f}_m\right) \quad (3.15)$$

$$q_{PIC}\left(\boldsymbol{f}_*\mid\boldsymbol{f}_m\right)=\mathcal{N}\left(\boldsymbol{f}_*\mid\boldsymbol{k}_{*m}\boldsymbol{K}_{mm}^{-1}\boldsymbol{f}_m,\operatorname{blockdiag}\left[K_{**}-Q_{**}\right]\right) \quad (3.16)$$

Low 等（2015）[54]则尝试保留了更丰富的分块信息。

Snelson 和 Ghahramani（2006）[55]基于 FITC 考虑了如何选取诱导点位置的问题。虽说诱导点是隐变量，但最简单的诱导点选取为数据集的子集，当然在这里我们不考虑如何选取子集的具体方法，但这些方法大多都与模型的训练相独立，而文献中则提出了诱导点位置 $\boldsymbol{X}_m$ 与超参数联合优化的方法，突破了诱导点必须是训练子集的限制。

SMGP[56]（Sparse Multiscale GP）则对高斯核函数进行了拓展。使用 $\boldsymbol{u}$ 去替代诱导点 $\boldsymbol{f}_m$，此时的 $\boldsymbol{u}$ 为

$$\left(\left\langle f_1,f\right\rangle_{\mathcal{H}},\cdots,\left\langle f_m,f\right\rangle_{\mathcal{H}}\right)^{\top} \quad (3.17)$$

是关于 $k\left(\cdot,\cdot\right)=cg\left(\cdot,\cdot,\boldsymbol{\sigma}\right)$ 的再生核希尔伯特空间 $\mathcal{H}$ 上隐函数 f 和基函数 f_i 之间的内积向量，其中 $c>0$ 是常数，$\boldsymbol{\sigma}>\mathbf{0}\in\mathbb{R}^d$，基函数是高斯的，有

$$u_i\left(\boldsymbol{x}\right)=g\left(\boldsymbol{x},\boldsymbol{v}_i,\boldsymbol{\sigma}_i\right)=\left|2\pi\operatorname{diag}\left(\boldsymbol{\sigma}_i\right)\right|^{-1/2}\exp\left(-\frac{1}{2}\sum_{d=1}^{D}\frac{\left(\left[\boldsymbol{x}-\boldsymbol{v}_i\right]_d\right)^2}{\left[\boldsymbol{\sigma}_i\right]_d}\right) \quad (3.18)$$

其中 $\left\{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_m\right\}$ 为训练集的子集。进一步有训练条件概率的近似

$$q_{SMGP}\left(\boldsymbol{f}\mid\boldsymbol{u}\right)=\mathcal{N}\left(\boldsymbol{f}\mid\boldsymbol{U}_{nm}\boldsymbol{U}_{\Psi}^{-1}\boldsymbol{u},\operatorname{diag}\left[\boldsymbol{K}_{nn}-\boldsymbol{U}_{nm}\boldsymbol{U}_{\Psi}^{-1}\boldsymbol{U}_{mn}\right]\right) \quad (3.19)$$

其中 $\left[\boldsymbol{U}_{mn}\right]_{i,j}=g\left(\boldsymbol{x}_j,\boldsymbol{v}_i,\boldsymbol{\sigma}_i\right)$，$\left[\boldsymbol{U}_{\Psi}\right]_{i,j}=g\left(\boldsymbol{v}_i,\boldsymbol{v}_j,\boldsymbol{\sigma}_i+\boldsymbol{\sigma}_j-\boldsymbol{\sigma}\right)/c$。该方法基于基函数从单一高斯分布到混合高斯分布的拓展，若取基函数为 $f_i=k\left(\boldsymbol{v}_i,\cdot\right)$，则 $\boldsymbol{u}$ 简化为 $\left(f\left(\boldsymbol{v}_1\right),\ldots,f\left(\boldsymbol{v}_m\right)\right)^{\top}$，且 $\boldsymbol{U}_{\Psi}$ 和 $\boldsymbol{U}_{mn}$ 简化为 $\boldsymbol{K}_{mm}$ 和 $\boldsymbol{K}_{mn}$。SMGP 是在 FITC 基础上使用 SEARD 核的拓展形式，其对于不同的基函数使用了不同的分布特征，即长度尺度（Length-Scales）。

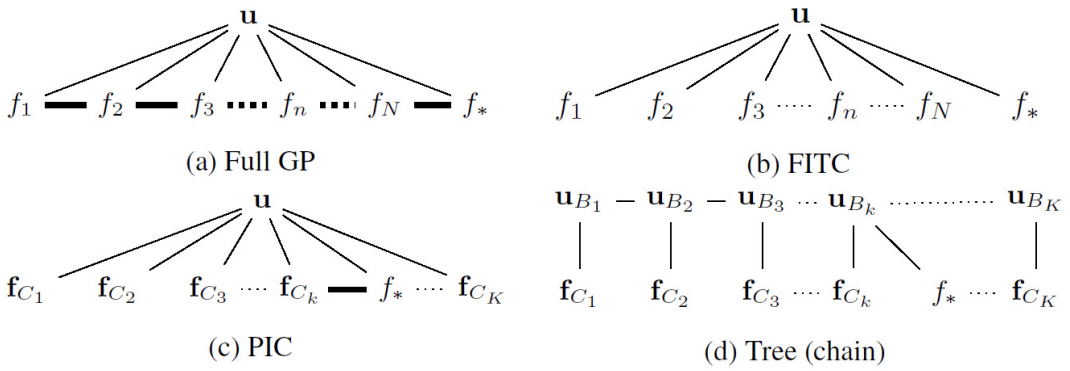

图 3.2　先验近似的图模型[57]。注：为加以区分，改图的诱导点使用符合 $\boldsymbol{u}$，而不是 $\boldsymbol{f}_m$。

PITC 将 $\boldsymbol{f}$ 分为 K 个不相交的区域 $\left\{\boldsymbol{f}_C^k\right\}_{k=1}^K$，则 TGP（Tree-structured GP）[57] 则进一步将诱导点 $\boldsymbol{f}_m$ 进行分区 $\left\{\boldsymbol{f}_{m,B}^k\right\}_{k=1}^K$。为了近似联合先验 $p\left(\boldsymbol{f},\boldsymbol{f}_m\right)\approx q\left(\boldsymbol{f}\mid\boldsymbol{f}_m\right)q\left(\boldsymbol{f}_m\right)$，则诱导点的先验和似然可以近似为

$$q\left(\boldsymbol{f}_m\right)=\prod_{k=1}^{K}p\left(\boldsymbol{f}_{m,B}^k\mid\boldsymbol{f}_{m,par(B)}^k\right) \tag{3.20}$$

$$q\left(\boldsymbol{f}\mid\boldsymbol{f}_m\right)=\prod_{k=1}^{K}p\left(\boldsymbol{f}_C^k\mid\boldsymbol{f}_{m,B}^k\right) \tag{3.21}$$

其中 $\boldsymbol{f}_{m,par(B)}^k$ 表示为 B 分区中第 k 个分区的父节点上的诱导点，且在协方差中记 $\boldsymbol{f}_C^k=c$，$\boldsymbol{f}_{m,B}^k=b$，$\boldsymbol{f}_{m,par(B)}^k=p$，则条件分布可得

$$p\left(\boldsymbol{f}_{m,B}^k\mid\boldsymbol{f}_{m,par(B)}^k\right)=\mathcal{N}\left(\boldsymbol{f}_{m,B}^k\mid\boldsymbol{K}_{bp}\boldsymbol{K}_{pp}^{-1}\boldsymbol{f}_{m,par(B)}^k,\boldsymbol{K}_{bb}-\boldsymbol{K}_{bp}\boldsymbol{K}_{pp}^{-1}\boldsymbol{K}_{pb}\right), \tag{3.22}$$

$$p\left(\boldsymbol{f}_C^k\mid\boldsymbol{f}_{m,B}^k\right)=\mathcal{N}\left(\boldsymbol{f}_C^k\mid\boldsymbol{K}_{cb}\boldsymbol{K}_{bb}^{-1}\boldsymbol{f}_{m,B}^k,\boldsymbol{K}_{cc}-\boldsymbol{K}_{cb}\boldsymbol{K}_{bb}^{-1}\boldsymbol{K}_{bc}\right) \tag{3.23}$$

TGP 的提出源于这样一个现象：诱导点只能刻画输入域中很小一部分范围的信息，但在计算中我们把它当作是全局的近似。所以，TGP 更类似于局部近似，但此处作为该框架下自然的延伸，将其归到该类。当然，当我们有足够的诱导点时则不存在上述问题或者说是现象。

3.1.1.2 谱近似方法

与传统的诱导点方法不同的是，IDGP（Inter-Domain GP）[58]提出了更广义的特征分解方法，即首先将输入域变换到一个不同的域上。数据在这个新的域上可以是更紧致的，有利于函数的估计与识别，并且一个特殊的域也使得我们能够加入一些先验的知识，有效地分解与使用核函数。考虑 $\boldsymbol{x}\in\mathbb{R}^D$ 上的 GP 的值 $f\left(\boldsymbol{x}\right)$，给定确定性实函数 $g\left(\boldsymbol{x},\boldsymbol{z}\right)$ 有 $\boldsymbol{z}\in\mathbb{R}^H$，定义线性变换

$$u\left(\boldsymbol{z}\right)=\int_{\mathbb{R}^D}f\left(\boldsymbol{x}\right)g\left(\boldsymbol{x},\boldsymbol{z}\right)d\boldsymbol{x} \tag{3.24}$$

此时，$f\left(\boldsymbol{x}\right)$ 和 $u\left(\boldsymbol{z}\right)$ 可被当作是一个联合的 GP。假设 $f\left(\boldsymbol{x}\right)\sim\mathcal{GP}\left(\mu\left(\boldsymbol{x}\right),k\left(\boldsymbol{x},\boldsymbol{x}'\right)\right)$，则变化之后域上的统计特征与域间的统计特征可以表示如下

$$\mu\left(\boldsymbol{z}\right)=\int_{\mathbb{R}^D}\mu\left(\boldsymbol{x}\right)g\left(\boldsymbol{x},\boldsymbol{z}\right)d\boldsymbol{x} \tag{3.25}$$

$$k\left(\boldsymbol{z},\boldsymbol{z}'\right)=\int_{\mathbb{R}^D}\int_{\mathbb{R}^D}k\left(\boldsymbol{x},\boldsymbol{x}'\right)g\left(\boldsymbol{x},\boldsymbol{z}\right)g\left(\boldsymbol{x}',\boldsymbol{z}'\right)d\boldsymbol{x}d\boldsymbol{x}' \tag{3.26}$$

$$k\left(\boldsymbol{x},\boldsymbol{z}'\right)=\int_{\mathbb{R}^D}k\left(\boldsymbol{x},\boldsymbol{x}'\right)g\left(\boldsymbol{x}',\boldsymbol{z}'\right)d\boldsymbol{x}' \tag{3.27}$$

本文不讨论函数 $g\left(\boldsymbol{x},\boldsymbol{z}\right)$ 的具体形式，Figueirasvidal 和 Lázarogredilla（2009）[58] 间接使用傅里叶变换，并讨论了 IDGP 与 FITC 和 SMGP 模型的关系。

Lázaro-Gredilla 等（2010）[59]对于平稳核可以令 $\boldsymbol{\tau}=\boldsymbol{x}-\boldsymbol{x}'$ 简化核函数的表示 $k\left(\boldsymbol{x},\boldsymbol{x}'\right)=k\left(\boldsymbol{\tau}\right)$，且维纳-辛钦定理说明这种随机过程下的协方差函数与谱密度 $S\left(\boldsymbol{s}\right)$ 是一个傅里叶变换对

$$k(\boldsymbol{\tau})=\int S(\boldsymbol{s})e^{2\pi i\boldsymbol{s}^{\top}\boldsymbol{\tau}}d\boldsymbol{s} \tag{3.28}$$

$$S(\boldsymbol{s})=\int k(\boldsymbol{\tau})e^{-2\pi i\boldsymbol{s}^{\top}\boldsymbol{\tau}}d\boldsymbol{\tau} \tag{3.29}$$

注意随机过程的方差为$k(\mathbf{0})=\int S(\boldsymbol{s})d\boldsymbol{s}$，以 SEARD 核函数为例

$$k_{SEARD}(\boldsymbol{\tau})=\sigma_{out}^2\exp\left(-\frac{1}{2}\boldsymbol{\tau}^{\top}\Lambda^{-1}\boldsymbol{\tau}\right), \tag{3.30}$$

其中$\Lambda=\mathrm{diag}\left(\left[l_1^2,\ldots,l_D^2\right]\right)$，超参数$\sigma_{out}^2$为先验方差，也被称为输出信号，$\left\{l_d^2\right\}$为输入特征的长度尺度，它决定了协方差函数随着距离衰减的程度，那么有$k(\mathbf{0})=\sigma_{out}^2$。根据博赫纳定理，有谱密度正比于概率测度$p_S(\boldsymbol{s})=\sqrt{|2\pi\Lambda|}\exp\left(-2\pi^2\boldsymbol{s}^{\top}\Lambda\boldsymbol{s}\right)$，则有

$$k(\boldsymbol{x},\boldsymbol{x}')=\sigma_{out}^2\mathbb{E}_{\boldsymbol{s}\sim p_S(\boldsymbol{s})}\left[\cos\left(2\pi\boldsymbol{s}^{\top}(\boldsymbol{x}-\boldsymbol{x}')\right)\right] \tag{3.31}$$

此协方差可使用蒙特卡洛方法近似，有$k(\boldsymbol{x},\boldsymbol{x}')\approx\sigma_{out}^2\sum_{i=1}^{m}\cos\left(2\pi\boldsymbol{s}_i^{\top}(\boldsymbol{x}-\boldsymbol{x}')\right)\Big/m$。

Hoang 等（2020）[60]则进一步考虑将先验$f(\boldsymbol{x})$由 p 个基函数近似$f'(\boldsymbol{x})=\sum_{i=1}^{p}f_i(\boldsymbol{x})$，其中$f_i(\boldsymbol{x})\sim\mathcal{GP}\left(0,\left(1/\sqrt{p}\right)k_i(\boldsymbol{x},\boldsymbol{x}')\right)$。其改进方向与 SMGP 类似，并且文章给出了近似的理论保证。

Wilson 和 Adams（2013）[61]基于对 SE 核谱密度是以原点为中心的高斯谱密度的观察，其无法囊括所有平稳核的表现。考虑一维的单高斯分布

$$\varphi(s;\mu,\sigma^2)=\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{1}{2\sigma^2}(s-\mu)^2\right], \tag{3.32}$$

令$S(s)=\left[\varphi(s)+\varphi(-s)\right]/2$带回傅里叶变换可得$k(\tau)=\exp\left(-2\pi^2\tau^2\sigma^2\right)\cos(2\pi\tau\mu)$。则当$\varphi(\boldsymbol{s})$是$\mathbb{R}^D$上的 Q 个混合高斯分布，其中第 q 个分量的均值与方差分别为$\boldsymbol{\mu}_q=\left(\mu_q^{(1)},\ldots,\mu_q^{(D)}\right)$和$\boldsymbol{M}_q=diag\left(v_q^{(1)},\ldots,v_q^{(D)}\right)$，则核函数可以记为

$$k(\boldsymbol{\tau})=\sum_{q=1}^{Q}w_q\prod_{d}^{D}\exp\left(-2\pi^2\tau_d^2v_q^{(d)}\right)\cos\left(2\pi\tau_d\mu_q^{(d)}\right) \tag{3.33}$$

其中w_q是描述第 q 个高斯分布贡献的权重。

Rahimi 和 Recht（2007）[62]将输入数据通过随机映射$\boldsymbol{z}:\mathbb{R}^D\to\mathbb{R}^M$到随机的低维特征空间中，则可以使用该映射近似核函数

$$k(\boldsymbol{x},\boldsymbol{x}')=\left\langle\boldsymbol{\psi}(\boldsymbol{x}),\boldsymbol{\psi}(\boldsymbol{x}')\right\rangle\approx\boldsymbol{z}(\boldsymbol{x})^{\top}\boldsymbol{z}(\boldsymbol{x}') \tag{3.34}$$

其中$\boldsymbol{\psi}$与映射至的特征空间的维度等于样本量的映射相关，M 维相较之下是低维的。给定核函数通过傅里叶变换得的概率测度$p_S(\boldsymbol{s})=\frac{1}{2\pi}\int e^{-i\boldsymbol{s}^{\top}\boldsymbol{\tau}}k(\boldsymbol{\tau})d\boldsymbol{\tau}$，独立同分布从$p_S$抽样 M 个样本$\boldsymbol{s}_1,\ldots,\boldsymbol{s}_M\in\mathbb{R}^D$及均匀分布$U[0,2\pi]$上抽样$b_1,\ldots,b_M\in\mathbb{R}$。令

$$\boldsymbol{z}(\boldsymbol{x})=\sqrt{\frac{2}{D}}\left[\cos\left(\boldsymbol{s}_1^{\top}\boldsymbol{x}+b_1\right),\ldots,\cos\left(\boldsymbol{s}_M^{\top}\boldsymbol{x}+b_M\right)\right]^{\top} \tag{3.35}$$

则可得到核函数的无偏估计。文章同时给出了一些典型核函数的概率测度，并给出了近似的收敛性。

Solin 和 Särkkä（2019）[63]根据拉普拉斯算子在紧致子集$\Omega \subset \mathbb{R}^D$中的特征函数展开获得核函数的近似特征分解。定义协方差算子$\mathcal{K}\phi = \int k(\cdot, \boldsymbol{x}')\phi(\boldsymbol{x}')d\boldsymbol{x}'$，可得到级数形式

$$\mathcal{K} = a_0 + a_1\left(-\nabla^2\right) + a_2\left(-\nabla^2\right)^2 + a_3\left(-\nabla^2\right)^3 + \cdots \tag{3.36}$$

其中∇^2为拉普拉斯算子。进而可以得到核函数的近似

$$k(\boldsymbol{x}, \boldsymbol{x}') \approx \sum_j S\left(\sqrt{\lambda_j}\right)\phi_j(\boldsymbol{x})\phi_j(\boldsymbol{x}') \tag{3.37}$$

其中$S(\cdot)$为协方差函数的谱密度，λ_j为拉普拉斯算子在Ω上第 j 个特征值并且$\phi_j(\cdot)$为相应的特征函数。

Bengio 等（2003）[64]使用傅里叶分解性质，关于再生核希尔伯特空间范数（$\mathcal{H}$范数）的最优核函数近似是如下展开形式

$$k(\boldsymbol{x}, \boldsymbol{x}') \approx \sum_{i=1}^{m} \lambda_i \phi_i(\boldsymbol{x})\phi_i(\boldsymbol{x}') \tag{3.38}$$

其中(λ_i, ϕ_i)为 m 个特征值最大所对应的特征值-特征函数对。

Wilson 等（2015）[65]说明 Kronecker 和 Toeplitz 方法受制于输入必须是网格形式的（如时间序列），其中 Kroncker 结果有利于核矩阵的特征分解但只适用于多维输入的情况，而 Toeplitz 方法能加速 MVM 的运算但只适用于一维。如果隐变量是在规则间隔的多维网格上的，并且核函数为平稳的乘性核（如 SE 核），那么结构核插值（structured kernel approximation）近似的$\boldsymbol{K}_{mm}$可以分解为 Toeplitz 矩阵$\boldsymbol{T}_i$的 Kroncker 乘积$\otimes$形式

$$\boldsymbol{K}_{mm} = \boldsymbol{T}_1 \otimes \cdots \otimes \boldsymbol{T}_P \tag{3.39}$$

进一步，$m \times m$的 Toeplitz 矩阵可以使用$a \times a$的 circulant 矩阵$\boldsymbol{C}$加速运算，因为其具有有效的特征分解形式

$$\boldsymbol{C} = \boldsymbol{F}^{-1}\mathrm{diag}(\boldsymbol{F}\boldsymbol{c})\boldsymbol{F} \tag{3.40}$$

其中$\boldsymbol{c} = [c_1, c_2, \ldots, c_2, c_1]^\top$，$C_{i,j} = \boldsymbol{c}_{|j-i| \bmod a}$，$F_{jk} = \exp(-2jk\pi i/a)$是离散傅里叶变换

3.1.2 后验稀疏近似

全局近似的第二部分被称为“精确先验，近似后验”，一般使用变分推断方法近似对数边际似然$\log p(\boldsymbol{y})$来训练（选择）模型的超参数，其使用变分分布$q(\boldsymbol{f}, \boldsymbol{f}_m)$去近似后验分布$p(\boldsymbol{f}, \boldsymbol{f}_m \mid \boldsymbol{y})$，但对于模型的加速，一般体现在诱导点的使用上。回顾诱导点方法的模型设置

$$p(\boldsymbol{f} \mid \boldsymbol{f}_m) = \mathcal{N}\left(\boldsymbol{f} \mid \boldsymbol{K}_{nm}\boldsymbol{K}_{mm}^{-1}\boldsymbol{f}_m, \tilde{\boldsymbol{K}}\right), \tag{3.41}$$

$$p(\boldsymbol{f}_m)=\mathcal{N}(\boldsymbol{f}_m\mid\boldsymbol{K}_{mm}) \tag{3.42}$$

其中 $\tilde{\boldsymbol{K}}=\boldsymbol{K}_{nn}-\boldsymbol{K}_{nm}\boldsymbol{K}_{mm}^{-1}\boldsymbol{K}_{mn}$。首先使用琴生不等式，可以得到

$$\log p(\boldsymbol{y}\mid\boldsymbol{f}_m)=\log\mathbb{E}_{p(\boldsymbol{f}\mid\boldsymbol{f}_m)}p(\boldsymbol{y}\mid\boldsymbol{f})\geq\mathbb{E}_{p(\boldsymbol{f}\mid\boldsymbol{f}_m)}\log p(\boldsymbol{y}\mid\boldsymbol{f})\triangleq\mathcal{L}_1 \tag{3.43}$$

此时的下界 $\mathcal{L}_1$ 与 $\log p(\boldsymbol{y})$ 的差异可以用 $KL\left(q(\boldsymbol{f}\mid\boldsymbol{f}_m)\|p(\boldsymbol{f}\mid\boldsymbol{f}_m,\boldsymbol{y})\right)$ 表示。接着继续对对数边际似然函数使用琴生不等式，可以进一步得到一个下界

$$\log p(\boldsymbol{y})=\log\int p(\boldsymbol{y}\mid\boldsymbol{f}_m)p(\boldsymbol{f}_m)d\boldsymbol{f}_m\geq\log\int\exp\{\mathcal{L}_1\}p(\boldsymbol{f}_m)d\boldsymbol{f}_m\triangleq\mathcal{L}_2 \tag{3.44}$$

值得注意的是，该结果匹配 Titsias（2009）[66]。其使用 $q(\boldsymbol{f},\boldsymbol{f}_m)=p(\boldsymbol{f}\mid\boldsymbol{f}_m)q(\boldsymbol{f}_m)$ 近似真实的分布 $p(\boldsymbol{f},\boldsymbol{f}_m\mid\boldsymbol{y})=p(\boldsymbol{f}\mid\boldsymbol{f}_m,\boldsymbol{y})p(\boldsymbol{f}_m\mid\boldsymbol{y})$，其中 $q(\boldsymbol{f}_m)$ 是变分分布，例如使用高斯分布 $q(\boldsymbol{f}_m)=\mathcal{N}(\boldsymbol{f}_m\mid\boldsymbol{m},\boldsymbol{S})$。并且由 $\boldsymbol{f}_m$ 是 $\boldsymbol{f}$ 的充分统计量的性质得，有 $p(\boldsymbol{f}\mid\boldsymbol{f}_m,\boldsymbol{y})=p(\boldsymbol{f}\mid\boldsymbol{f}_m)$。此时，最小化散度 $KL\left(q(\boldsymbol{f},\boldsymbol{f}_m)\|p(\boldsymbol{f},\boldsymbol{f}_m\mid\boldsymbol{y})\right)$ 等价于最大化变分下界

$$F_V\left(\boldsymbol{X}_m,q(\boldsymbol{f}_m)\right)=\log p(\boldsymbol{y})-KL(q\|p)=\int_{\boldsymbol{f},\boldsymbol{f}_m}q(\boldsymbol{f},\boldsymbol{f}_m)\log\frac{p(\boldsymbol{y},\boldsymbol{f},\boldsymbol{f}_m)}{q(\boldsymbol{f},\boldsymbol{f}_m)}d\boldsymbol{f}d\boldsymbol{f}_m\triangleq\mathcal{L}_3 \tag{3.45}$$

其中，可以联合优化 $\boldsymbol{X}_m$ 和 $q(\boldsymbol{f}_m)$。此时，经过变形（细节见 Titsias（2009）[67]）得

$$F_V\left(\boldsymbol{X}_m,q(\boldsymbol{f}_m)\right)=\int_{\boldsymbol{f}_m}q(\boldsymbol{f}_m)\left\{\log\frac{\exp(\mathcal{L}_1)p(\boldsymbol{f}_m)}{q(\boldsymbol{f}_m)}\right\}d\boldsymbol{f}_m \tag{3.46}$$

对 F_V 使用琴生不等式，或令其对于 $q(\boldsymbol{f}_m)$ 的导数为零，都可以得到一个更紧的下界

$$F_V\left(\boldsymbol{X}_m\right)=\log\int_{\boldsymbol{f}_m}\exp(\mathcal{L}_1)p(\boldsymbol{f}_m)d\boldsymbol{f}_m=\mathcal{L}_2 \tag{3.47}$$

有 $\mathcal{L}_2\geq\mathcal{L}_3$，因为此时变分分布已使用最优的形式 $q^*(\boldsymbol{f}_m)$，且只需优化诱导点的输入位置 $\boldsymbol{X}_m$。值得注意的是，该方法对诱导点的使用和预测分布与 DTC 相同。Hoang（2016）[68]也为变分法的稀疏 GP 模型提供了分布式训练的框架。

Hensman（2013）[71]则使用了 $\mathcal{L}_3$ 下界，可以重新写成

$$\mathcal{L}_3=\mathbb{E}_{p(\boldsymbol{f}\mid\boldsymbol{f}_m)q(\boldsymbol{f}_m)}\log p(\boldsymbol{y}\mid\boldsymbol{f})-KL\left(q(\boldsymbol{f}_m)\|p(\boldsymbol{f}_m)\right) \tag{3.48}$$

由于 $p(\boldsymbol{y}\mid\boldsymbol{f})=\prod_{i=1}^{n}p(y_i\mid f_i)$，则该下界可以写成关于输入样本对 $\{\boldsymbol{x}_i,y_i\}$ 的 n 项求和形式，此性质导出了有效的随机梯度方法。并且变分参数 $\boldsymbol{m}$ 及 $\boldsymbol{S}$ 被定义在非欧空间上，使用基于欧式距离的标准梯度方法难以有效地更新参数，故可以使用随机自然梯度下降法迭代更新 $\boldsymbol{m}$ 及 $\boldsymbol{S}$，细节见 Hensman（2013）[71]。

Hoang 等（2015）[72]将随机梯度方法应用于 DTC 以外的诱导点近似模型。变分方法与诱导点方法的联系可以由预测分布 $p(f_*\mid\boldsymbol{y})$ 建立

$$p(f_*|\boldsymbol{y})=\int p(f_*|\boldsymbol{f}_m,\boldsymbol{y})\,p(\boldsymbol{f}_m|\boldsymbol{y})\,d\boldsymbol{f}_m \simeq \int q(f_*|\boldsymbol{f}_m)\,q^*(\boldsymbol{f}_m)\,d\boldsymbol{f}_m \quad (3.49)$$

其中$q(f_*|\boldsymbol{f}_m)$为诱导点方法的测试条件分布，$q^*(\boldsymbol{f}_m)$为变分方法的最优变分分布，对应于不同的诱导点方法都有$q^*(\boldsymbol{f}_m)=p(\boldsymbol{f}_m|\boldsymbol{y})$。回顾 DTC 及其变分形式，输出的预测分布都是相同的，只是变分方法中用于训练的对数边际似然函数在形式上多了迹的项[67]。具体地，不同的诱导点方法使用了不同的训练条件分布$q(\boldsymbol{f}|\boldsymbol{f}_m)$，最优的变分分布可以得到$q^*(\boldsymbol{f}_m)=\mathcal{N}(\boldsymbol{\mu}^*,\boldsymbol{\Sigma}^*)$，其中

$$\boldsymbol{\mu}^*=\boldsymbol{K}_{mn}\left(\boldsymbol{Q}_{mm}+\boldsymbol{\Gamma}\right)^{-1}\boldsymbol{y} \quad (3.50)$$

$$\boldsymbol{\Sigma}^*=\boldsymbol{K}_{mm}-\boldsymbol{K}_{mn}\left(\boldsymbol{Q}_{mm}+\boldsymbol{\Gamma}\right)^{-1}\boldsymbol{K}_{nm} \quad (3.51)$$

$\boldsymbol{\Gamma}\triangleq\boldsymbol{R}+\sigma_\varepsilon^2\boldsymbol{I}$，对于 PIC 与 PITC 有$\boldsymbol{R}=\mathrm{blockdiag}[\boldsymbol{K}_{nn}-\boldsymbol{Q}_{nn}]$，对于 FIC 与 FITC 有$\boldsymbol{R}=\mathrm{diag}[\boldsymbol{K}_{nn}-\boldsymbol{Q}_{nn}]$，对于 DTC 与 SoR 有$\boldsymbol{R}=\boldsymbol{0}$。进一步，当$\boldsymbol{\Sigma}^{*-1}$与$\boldsymbol{\Sigma}^{*-1}\boldsymbol{\mu}^*$满足可分解性条件时，随机梯度是标准梯度的无偏估计。该可分解性条件可表示如下

$$\boldsymbol{\Sigma}^{*-1}=F'(\boldsymbol{f}_m)+\sum_{i=1}^{M}F\left(\boldsymbol{f}_m,\boldsymbol{y}_{\mathcal{D}_i}\right) \quad (3.52)$$

$$\boldsymbol{\Sigma}^{*-1}\boldsymbol{\mu}^*=G'(\boldsymbol{f}_m)+\sum_{i=1}^{M}G\left(\boldsymbol{f}_m,\boldsymbol{y}_{\mathcal{D}_i}\right) \quad (3.53)$$

其中原始数据集$\mathcal{D}$被划分为了 M 个不相交的区域$\mathcal{D}_i$，有$\boldsymbol{y}_{\mathcal{D}_i}\triangleq(y_{x'})_{x'\in\mathcal{D}_i}^{\top}$，$F$、$F'$、$G$与$G'$为关于括号内变量的任意函数。

Bui 等[73]后验概率$p(\boldsymbol{f}|\boldsymbol{y})$可以被近似为：

$$p(\boldsymbol{f}|\boldsymbol{y})=\frac{1}{p(\boldsymbol{y})}p(\boldsymbol{f})\prod_{i=1}^{n}p(y_i|f_i)\approx\frac{1}{Z_{PEP}}p(\boldsymbol{f})\prod_{i=1}^{n}t_i(\boldsymbol{f}_m)\triangleq q(\boldsymbol{f}) \quad (3.54)$$

其中精确的条件概率$p(y_i|f_i)$被假设是高斯的简单因子$t_i(\boldsymbol{f}_m)$所近似[17]。与变分法一样，$t_i(\boldsymbol{f}_m)$可由最小化$KL(p(\boldsymbol{f}|\boldsymbol{y})\|q(\boldsymbol{f}))$得。但直接全部求解所有的因子是困难的，所以在 EP（Expectation Propagation）算法中，每个$t_i(\boldsymbol{f}_m)$则是在每次迭代中分别近似$p(y_i|f_i)$。令非化标准分布为$\hat{q}(\boldsymbol{f})\triangleq p(\boldsymbol{f})\prod_{i=1}^{n}t_i(\boldsymbol{f}_m)$，首先，计算空心分布（Cavity Distribution）$q^{\backslash i}(\boldsymbol{f})\propto\hat{q}(\boldsymbol{f})/t_i^{\alpha}(\boldsymbol{f}_m)$。然后，更新非标准化的分布$\hat{q}(\boldsymbol{f})=\arg\min_{\hat{q}(\boldsymbol{f})}KL\left(q^{\backslash i}(\boldsymbol{f})\,p^{\alpha}(y_i|f_i)\|\hat{q}(\boldsymbol{f})\right)$。最后，更新后的因子可表示为$t_i(\boldsymbol{f}_m)=t_{i,old}^{1-\alpha}(\boldsymbol{f}_m)\,t_{i,new}^{\alpha}(\boldsymbol{f}_m)$，其中$t_{i,new}^{\alpha}(\boldsymbol{f}_m)=\hat{q}(\boldsymbol{f})/q^{\backslash i}(\boldsymbol{f})$。特别的，在高斯过程回归框架下，有最终结果：

$$t_i(\boldsymbol{f}_m)=\mathcal{N}\left(K_{im}K_{mm}^{-1}\boldsymbol{f}_m|\,y_i,\alpha K_{ii}+\sigma_\varepsilon^2\right),\quad q(\boldsymbol{f}_m)=\mathcal{N}\left(\boldsymbol{f}_m|\,K_{mn}\hat{K}_{nn}^{-1}\boldsymbol{y},K_{mm}-K_{mn}\hat{K}_{nn}^{-1}K_{nm}\right) \quad (3.55)$$

其中$\hat{K}_{nn}=Q_{nn}+\alpha\mathrm{diag}\left(\tilde{K}_{nn}\right)+\sigma_\varepsilon^2$。而且，近似的对数边际似然也有解析形式

$$\log Z_{PEP}=-\frac{N}{2}\log(2\pi)-\frac{1}{2}\log\left|\hat{K}_{nn}\right|-\frac{1}{2}\boldsymbol{y}^{\top}\hat{K}_{nn}\boldsymbol{y}+\frac{1-\alpha}{2\alpha}\sum_{n}\log\left(1+\alpha\tilde{K}_{ii}/\sigma_\varepsilon^2\right) \quad (3.56)$$

[17]可见 GPML[9]的第三章。

特别的，当$\alpha=1$时，Power EP 退化为 EP，近似后验$q(\boldsymbol{f}_m)$与边际似然$\log Z_{PEP}$同 FITC；当$\alpha\to 0$时，近似后验与边际似然的形式等同于 Titsias 的变分下界。

Hernandez-Lobato 等（2015）[75]随机 EP 方法则使用了不同的空心分布$q^{\backslash i}(\boldsymbol{f})\propto \hat{q}(\boldsymbol{f})/t_{total}^{1/n}(\boldsymbol{f}_m)$，其中有全局因子$t_{total}=\prod_{i=1}^{n}t_i$。

Matthews 等（2016）[76]假设在可测空间$(\Omega,\mathcal{F})$上有两个测度μ和η，则存在 Radon-Nikodyn 导数$d\mu/d\eta$，以及测度间的 KL 散度的定义：

$$KL(\mu\|\eta)=\int_{\Omega}\log\left\{\frac{d\mu}{d\eta}\right\}d\mu=\int_{\Omega}\mu\log\left\{\frac{d\mu}{d\eta}\right\}dm, \tag{3.57}$$

其中勒贝格测度 m 与μ和η都有关。特别的，考虑三种函数集$f:X\to\mathbb{R}$上的概率测度。第一个为假设高斯过程的先验测度P，第二个为假设稀疏高斯过程的近似测度Q，第三个为后验测度$\hat{P}$。首先可得 Radon-Nikodyn 导数

$$\frac{dQ}{d\hat{P}}(f)=\frac{q(f_{D\backslash Z},f_Z)}{p(f_{D\backslash Z},f_Z\,|\,Y)} \tag{3.58}$$

其中 p 和 q 是相应的密度，对应于有限维情况下的 Lebesgue 测度。则随机过程间的 KL 散度可以退化为 Titsias 的散度（具体见文献）

$$KL\left(Q\|\hat{P}\right)=\int_{\mathbb{R}^X}\log\left\{\frac{dQ}{d\hat{P}}\right\}dQ=\int_{\mathbb{R}^{Z\cup D}}q(f_{D\backslash Z},f_Z)\log\left\{\frac{q(f_{D\backslash Z},f_Z)}{p(f_{D\backslash Z},f_Z\,|\,Y)}\right\}dm_{Z\cup D}。 \tag{3.59}$$

Hensman 和 Matthews（2015）[77]考虑带超参数$\boldsymbol{\theta}$的表示形式，预测后验可被近似为

$$p(f_*\,|\,\boldsymbol{y})\approx q(f)=\iint p(f_*\,|\,\boldsymbol{f}_m,\boldsymbol{\theta})q(\boldsymbol{f}_m,\boldsymbol{\theta})d\boldsymbol{\theta}d\boldsymbol{f}_m \tag{3.60}$$

其中$q(\boldsymbol{f}_m,\boldsymbol{\theta})$为采样分布。具体的，最小化$KL\left(q(f_*,\boldsymbol{f},\boldsymbol{f}_m,\boldsymbol{\theta})\|\,p(f_*,\boldsymbol{f},\boldsymbol{f}_m,\boldsymbol{\theta}\,|\,\boldsymbol{y})\right)$，可得用于 MCMC 采样的最优分布

$$\log q^*(\boldsymbol{f}_m,\boldsymbol{\theta})=\mathbb{E}_{p(\boldsymbol{f}|\boldsymbol{f}_m,\boldsymbol{\theta})}\left[\log p(\boldsymbol{y}\,|\,\boldsymbol{f})\right]+\log p(\boldsymbol{f}_m\,|\,\boldsymbol{\theta})+\log p(\boldsymbol{\theta})-\log C, \tag{3.61}$$

其中 C 为常数，且最优分布不受限于具体的分布形式。

另外，Huggins 等（2019）[78]因为小的 KL 散度不能说明预测误差也很小，所以使用了 pF（preconditioned Fisher）散度作为替代。Hensman 等（2017）[79]、Gal 和 Turner（2015）[80]主要结合了变分稀疏方法与谱方法。Bauer 等（2016）[81]主要考虑了 FITC 与变分 DTC（Titsias）的关系。

3.1.3 非稀疏近似

直接加速高斯过程回归可以使用迭代的方法求解线性系统$\left(K+\sigma_\varepsilon^2 I\right)\boldsymbol{v}=\boldsymbol{y}$，如共轭梯度法[18][82][83]，等同于求解优化问题$\boldsymbol{v}^*=\arg\min_{\boldsymbol{v}}0.5\boldsymbol{v}^\top\left(K+\sigma_\varepsilon^2 I\right)\boldsymbol{v}-\boldsymbol{v}^\top\boldsymbol{y}$，

[18]广义的可考虑 Krylov 子空间方法，如 Pleiss 等（2020）[84]。

在$k \ll n$次迭代之后，时间复杂度可由$\mathcal{O}\left(n^3\right)$降为$\mathcal{O}\left(kn^2\right)$。在每次迭代中，都需要用到矩阵向量乘法（matrix-vector multiplication，MVM）有$\boldsymbol{a}=\left(K+\sigma_\varepsilon^2 I\right)\boldsymbol{v}$，且第 i 个分量可以写为$a_i=\sum_{j=1}^{n} k\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right)v_j$，近似该形式则被称为快速 MVM 近似方法[85]。并且，MVM 方法可以被用在其他同样具有加权求和形式的地方，如求解预测均值。

常见的快速 MVM 方法分为稀疏核矩阵、数据结构近似、核结构近似三类[86]，其中稀疏核矩阵指的是矩阵中包含很多的零元素，可以加速计算。但对于一般情况，即使使用紧支（compact support）核[87]，也难以达到稀疏矩阵的效果。而在特殊的情况，如时间序列数据时，该方法是有效的，如使用 Toeplitz 核结构[88]，且该结构可以使用基于傅里叶变换的方法所加速[89]。数据结构近似则将训练数据划分为数据子集，如使用 KD 树[90]，其指标集在此记为S_c，则线性组合部分可近似为

$$\sum_{j=1}^{n} k\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right)v_j \approx \sum_c\left[\sum_{j\in S_c} v_j\right]k_c \tag{3.62}$$

其中k_c为常数，因为此时看作$\boldsymbol{x}_i$到S_c上所有$\boldsymbol{x}_j$的距离是相等的。此外，也可不近似而直接使用分布式计算$\left(K+\sigma_\varepsilon^2 I\right)\boldsymbol{v}$，其每个分量为$\left(K_{j\in S_c,n}+\sigma_\varepsilon^2 I\right)\boldsymbol{v}$[91]。最后，核结构近似则对应于具体的核函数，如对应高斯核的快速高斯变换[92]，其基于多元赫米特/泰勒级数展开，还有基于泰勒展开与输入数据分区的马顿核形式[93]，和基于输入数据装箱的直方图交叉核形式。除上述具体的 MVM 方法，也有与诱导点方法结合的近似结构，称为结构核插值方法[88][94]，其在迭代时使用近似的核函数$K_{nn}\approx WK_{mm}W^{\top}$，其中$W$为$n\times m$的插值矩阵，若使用三次插值则每行只有 4 个非零项，即表示每个数据点只与最近的四个诱导点有关。

迭代方法与快速 MVM 方法主要会面临两个问题，一是如同稀疏近似中，多少个诱导点能够保持良好的预测，即迭代次数 k 如何选取的问题，二是迭代法通常被应用于预测均值的计算当作，而考虑预测方差时，则需要增加其他的计算量，如考虑一个新的线性系统[95]。故也有部分工作是通过分解矩阵来求解逆矩阵，标准的方法为 Cholesky 分解。Nguyen 等（2019）[96]使用了分布式的 Cholesky 分解；Dietrich 和 Newsam（1997）[97]则对 Toeplitz 矩阵进行了重构，之后可直接进行特征分解；Ambikasran 等（2016）[98]则将协方差阵分解为分块对角阵乘积的形式。此外，也有工作只关注与模型训练时的加速，如 Xu 等（2019）[99]使用分布式的交替方向乘子法（alternating direction method of multipliers）解决超参数的优化问题，其中核矩阵分块方法加速了计算；Chen 等（2020）[100]则对完整的高斯过程回归模型的训练步骤使用了随机梯度下降算法而不是全梯度，完整核矩阵求逆的工作被小块的核矩阵所替代，导致计算消耗降低，并且给出了统计上的理论保证，如随机梯度训练的超参数能收敛到全梯度的驻点。回顾本段的改

进方法，大量使用了矩阵分块的方法，这种降低计算量的手段与局部方法类似。更加细化的，Almosallam 等（2016）[101]对于 SE 核函数求导结果给出了更高效的表现形式。

3.2 局部近似

局部近似的主要思想是“分而治之”，即将完整的数据集$\mathcal{D}$划分为 M 个数据子集$\{\mathcal{D}_i\}_{i=1}^{M}$，其中可用每个数据子集训练一个子 GP 模型[19]，最终的预测结果则是建立在子模型之上。本文将该部分模型分为三类：最近邻（nearest neighbor）模型、聚合（aggregation）模型、混合专家（mixture of experts）模型。根据如何使用子模型，可以将其分为最近邻模型与聚合模型，明显的，最近邻模型指的是预测时，只考虑与预测点较近的子模型给出预测，通常会造成连续域上预测的不连续性。而聚合模型则考虑了绝大多数子模型的预测结果，平滑了两个数据子集间的不连续预测，但依然存在 Kolmogorov 不一致的问题[105]。第三类的混合专家模型则与如何划分数据子集有关系，与前两者直接划分不同的是，其根据隐变量推断样本点属于哪个数据子集或子模型，如使用混合高斯分布。总而言之，局部近似方法使得模型可以通过分布式/并行运算加速，并且，子模型有利于识别如非平稳性、异方差性等局部特征。

3.2.1 最近邻模型

显而易见的是，最简单的最近邻模型只需选择离预测点最近的子模型作出预测[106]，而简单的划分区域方式可以是聚类算法或树结构[107]。但此方法很容易导致分区/临域间的不连续性，Park 等（2011）[108]以 Kriging 视角，在求解优化问题时，在临域间添加了约束性条件；Park 和 Apley（2018）[110]为了保持连续性，在临域间定义了新的随机过程；Urtasun 和 Darrell（2008）[111]考虑了最近几个子模型预测分布的加权平均，同样在部分程度上缓解了不连续性问题。

考虑最近的子模型的思想可以表现为马尔科夫性质，但当使用的子模型增加时，不得不添加独立性假设使得分布可以分解为子模型分布的乘积形式，如边际似然在 M 个子模型上的完全分解形式$p(\boldsymbol{y})\approx\prod_{i=1}^{M}p(\boldsymbol{y}_i)$。Moore 和 Russell（2015）[112]使用了无向图(G,E)缓解了独立性假设，其中 G 为顶点集，$E=\{(i,j)\,|\,1\le i<j\le M\}$表示边界集，将边际似然分解为

$$p(\boldsymbol{y})\approx\prod_{i=1}^{M}p(\boldsymbol{y}_i)\prod_{(i,j)\in E}\frac{p(\boldsymbol{y}_i,\boldsymbol{y}_j)}{p(\boldsymbol{y}_i)p(\boldsymbol{y}_j)},\tag{3.63}$$

其有相应的马尔科夫随机场形式。Lindgren 等（2011）[48]则考虑了当核函数为马

[19] 英文文献中常被称作 expert。

顿核的情况，可使用高斯马尔科夫随机场来表示高斯场，加速了数值计算的过程。

3.2.2 聚合模型

聚合模型虽说与之前的最近邻模型没有显著的差异，但不同之处在于：对于每一个预测点，聚合模型使用了大量的训练集数据（或大多子模型），而不是仅使用单个子模型或最近几个。可以看出，聚合模型以损失运算能力为代价，获得了更精确、鲁棒的预测结果，也同时防止了过拟合问题。对于分好区的 M 个数据子集 $\mathcal{D}_i$[20]，每个子模型是一个高斯过程回归模型，有预测分布 $p_i\left(y_*\mid\mathcal{D}_i,x_*\right)$，则最终的预测分布 $p\left(y_*\mid\mathcal{D},x_*\right)$ 由子模型的预测分布 “聚合”而得。简单平均有预测 $p\left(y_*\mid\mathcal{D},x_*\right)=\frac{1}{M}\sum_{i=1}^{M}p_i\left(y_*\mid\mathcal{D}_i,x_*\right)$[114]；Gx 行 PoE（generalized product of experts）[116]有预测 $p\left(y_*\mid\mathcal{D},x_*\right)=\prod_{i=1}^{M}p_i^{\beta_i}\left(y_*\mid\mathcal{D}_i,x_*\right)$，其中 β_i 为第 i 个子模型的贡献（或权重），当 $\beta_i=1$ 时退化为 PoE[117]；RBCM（robust Bayesian committee machine）[118]有预测

$$p\left(y_*\mid\mathcal{D},x_*\right)=\frac{\prod_{i=1}^{M}p_i^{\beta_i}\left(y_*\mid\mathcal{D}_i,x_*\right)}{p^{-1+\sum_i\beta_i}\left(y_*\mid x_*\right)},\tag{3.64}$$

当 $\beta_i=1$ 时退化为 BCM[119]，而 GRBCM（generalized RBCM）[120]则使用了从原始数据随机抽样的“全局”子集 $\mathcal{D}_c$[21]及增广数据子集 $\mathcal{D}_{+i}=\left\{\mathcal{D}_c,\mathcal{D}_i\right\}$，有

$$p\left(y_*\mid\mathcal{D},x_*\right)=\frac{\prod_{i=1}^{M}p_i^{\beta_i}\left(y_*\mid\mathcal{D}_{+i},x_*\right)}{p_c^{-1+\sum_i\beta_i}\left(y_*\mid\mathcal{D}_c,x_*\right)}。\tag{3.65}$$

NPAE（nested pointwise aggregation of experts）[122]则放宽了独立性假设，将预测的均值视为随机变量，因此可考虑不同子模型预测间的协方差，其最终预测的分布具有趋于完整 GP 回归模型预测分布的一致性[123]。

此外，也有许多小技巧可提升模型整体的预测能力或降低计算消耗，如使用增广的数据子集 $\left\{\mathcal{D}_i,\mathcal{D}_{i+1}\right\}$ 代替 $\mathcal{D}_i$[124]。刘晓芳等（2019）[127]也使用了重叠部分的信息，以 $\mathcal{D}_1\cup\mathcal{D}_3$ 与 $\mathcal{D}_2\cup\mathcal{D}_3$ 两个区域在 $\mathcal{D}_3$ 上重叠为例，其预测分布可近似为

$$p\left(y_*\mid\mathcal{D},x_*\right)=\frac{p\left(y_*\mid\mathcal{D}_1,\mathcal{D}_3,x_*\right)p\left(y_*\mid\mathcal{D}_2,\mathcal{D}_3,x_*\right)}{p\left(y_*\mid\mathcal{D}_3,x_*\right)}。\tag{3.66}$$

Gao 等（2020）[126]使用 Tsallis 互信息权重 $\beta_i=S_q\left(y_*\mid\boldsymbol{x}_*\right)-S_q\left(y_*\mid\boldsymbol{x}_*,\mathcal{D}_i\right)$ 替代常用的香农互信息 $\beta_i=H\left(y_*\mid\boldsymbol{x}_*\right)-H\left(y_*\mid\boldsymbol{x}_*,\mathcal{D}_i\right)$ 或等价权重 $\beta_i=1/M$，其中 $S_q\left(\cdot\right)$ 为 Tsallis 熵，$H\left(\cdot\right)$ 为香农熵。Li 等（2021）[121]则提出了度量满足独立性假设程度的指标，以点 $\boldsymbol{x}_*$ 落入 $\mathcal{D}_i$ 为例，有 $\gamma_*=\beta_i-\sum_{j\in KNN(*)}\beta_j$，其中 $KNN\left(*\right)$ 表示点 $\boldsymbol{x}_*$ 周围 K 个最近的数据子集。由 β_i 可知，当 $\mathcal{D}_i$ 提供的信息越多时，β_i 越大，而当周围的子集都提供很小的香农互信息时，γ_* 则很大，说明该点 $\boldsymbol{x}_*$ 满足假设

[20]聚合模型中数据子集的选取也可使用自助法[113]。
[21]该子集也可以使用诱导点[121]。

的程度越高。当除去数据集中部分γ很小的点时，即保证了模型的假设，也加速了运算。

3.2.3 混合专家模型

混合专家模型[128]一般被用来解决多任务回归问题，有简化形式$p\left(y_*\mid\mathcal{D},x_*\right)=\sum_{i=1}^{M}G_i\left(\boldsymbol{x}_*\right)p_i\left(y_*\mid\mathcal{D}_i,x_*\right)$，其中$G_i(\cdot)$被称为门控网络，控制预测时分配给不同子模型的概率，有$\sum_{i=1}^{M}G_i\left(\boldsymbol{x}_*\right)=1$。本部分局部近似的内容都可以归为广义的混合专家模型，但在此本文主要考虑子模型（或分区）是如何通过概率模型生成的狭义的情况。

一般的，高斯过程回归中的混合专家模型使用概率分布来代替门控网络，如狄利克雷分布[129]、波利亚分布[130]，此时，为了构建概率模型，会自然地将一些超参数也附加概率形式，如给定噪声超参数σ_ε^2一个逆伽马分布的先验。故混合专家模型与其他局部近似模型的不同之处在于，其超参数是由数据推断的而不是固定的，并且，使用狄利克雷分布可以有效地确定需要的子模型的个数。显然，在计算中添加了大量的概率分布，模型的推断便成了问题，MCMC（Markov Chain Monte Carlo）采样是常用的方法，但对于大数据而言仍显不足，Nguyen 等（2016）[131]则使用了变分法去处理该问题。此外，也有一些模型更加类似于聚合模型，如 Nguyen 和 Bonilla（2014）[132]虽然使用了概率先验形式的门控网络，但在预测时还是根据马氏距离去分配具体的子模型；Shi 等（2005）[133]也不类似最近邻模型使用单个子模型，而是如聚合模型使用多个子模型进行预测。最后，对于实时预测的任务，Nguyen-Tuong（2008）[134]也给出了迭代更新（或修正）分区的方法。

3.3 改变模型的方法

最后，涉及一些模型结构上变化导致计算复杂度下降的方法。如之前所提的神经过程，其已不再是高斯过程回归的框架，但确实降低了时间复杂度。Raissi（2017）[102]则将非参数高斯过程回归模型重新参数化，使用这些参数去编码训练集所获得的信息，使得预测时只需使用这些参数而不是原始的数据集。Sarkka 等（2013）[103]则将高斯过程回归模型的求解与状态空间模型（或随机微分方程）联系起来，即将特定核函数的高斯过程回归模型变换为等价的状态空间模型，然后使用卡尔曼滤波器求解，因为使用卡尔曼滤波器处理相应的问题只需要花费线性的时间复杂度。在时间花费上，该方法最具吸引力。

3.4 总结

至此，可以比较容易看出以诱导点为主的全局近似方法与“分而治之”的局部近似方法的缺陷。使用诱导点近似时，少量的全局隐变量难以刻画局部的数据特征，如非平稳性与异方差性。相反，局部近似方法的强项就在于捕捉数据的局部特征，但由于独立性假设，其相应损失了全局信息，也更容易过拟合。则有一部分工作是建立在融合两者特点的基础上，如之前所提的 PIC 即是从全局近似出发的，融合了分区的方法；Li 等（2021）[121]则从局部近似出发，融合了诱导点方法。Nguyen 等（2018）[135]则构建了一个双层的 GP 模型，其中顶层使用了全局近似，由诱导点建模全局的信息，而底层使用了局部近似，充分发掘数据的局部信息。

无论如何，这些近似方法通通都会涉及一个问题，即它的近似能力怎么样？更加具体的，对于稀疏近似方法，我们需要多少个诱导点或多少阶低秩可以近似完整的 GP 回归模型；对于非稀疏近似方法，我们需要迭代多少次才可以近似完整的 GP 回归模型；对于局部近似方法，我们需要多少个子模型或者说子模型需要包含多少个样本点才能使我们精确地近似完整的 GP 模型？抛开数学上的约束，这些问题等于使用者在预测精度以及时间复杂度上进行权衡，但我们总希望有更加精确的数学理论予以指导或者说对模型进一步控制。对于 GP 模型的低秩近似，Trecate（1999）[137]给出了建议，如选择核矩阵分解后特征值远大于 σ_ε^2/n 的分量的数量。并且，Dereziński 等（2020）[139]更加细化地考虑了低秩近似与使用数据子集间的关系。Burt 等（2019）[140]则关注使用诱导点加变分法的稀疏 GP 模型，通过为似然函数添加上下界的办法，给出建议，如对于 SE 核与 D 维输入，有建议的诱导点数量 $m=\mathcal{O}\left(\log^D n\right)$。同样的，Raykar 和 Duraiswami（2007）[141]在设计新的迭代方法的同时，给出了对迭代精度的分析，Das 等（2018）[113]则分别依据理论与经验给出了数据子集大小的选取建议。

当我们从理论上得知全局近似需要多少个诱导点或者局部近似中子集的大小如何时，在实际的应用中，如何选取诱导点或者如何为局部近似划分区域也会造成影响。局部近似而言，该部分的工作较少，因为分区的形式并不能直接体现在贝叶斯推断的过程中。对于混合专家模型，可以依赖狄利克雷过程生成分区，对于聚合模型，Liu 等（2018）[120]也考虑了随机分区与聚类方法导致的差异。全局近似方面的工作则相对较多，多基于原诱导点集 U 与添加新的诱导点后的集合 $U\cup\{\boldsymbol{x}_i,y_i\}$ 的指标进行比较，如：Smola 和 Bartlett（2000）[142]基于类似 Kriging 的角度，根据最小化预测的均方误差选择诱导点；Gramacy 和 Apley（2015）[143]则将训练集上的诱导点作为估计的对象，根据估计的均方误差选择诱导点；Schreiter 等（2015）[144]则根据训练集上的估计误差选择诱导点；Herbrich 等（2002）

[145]根据香农熵的差选择诱导点；Seeger 等（2003）[146]根据信息增益选择诱导点。并且，很大程度上选点的工作是根据先验的，但我们知道，稀疏近似的效果也在很大程度上取决于协方差的超参数，Titsias（2009）[67]也给出了一边根据一定的规则添加诱导点一边训练超参数的 EM 算法框架。

4 具有一致性的双层聚合模型

4.1 聚合模型框架

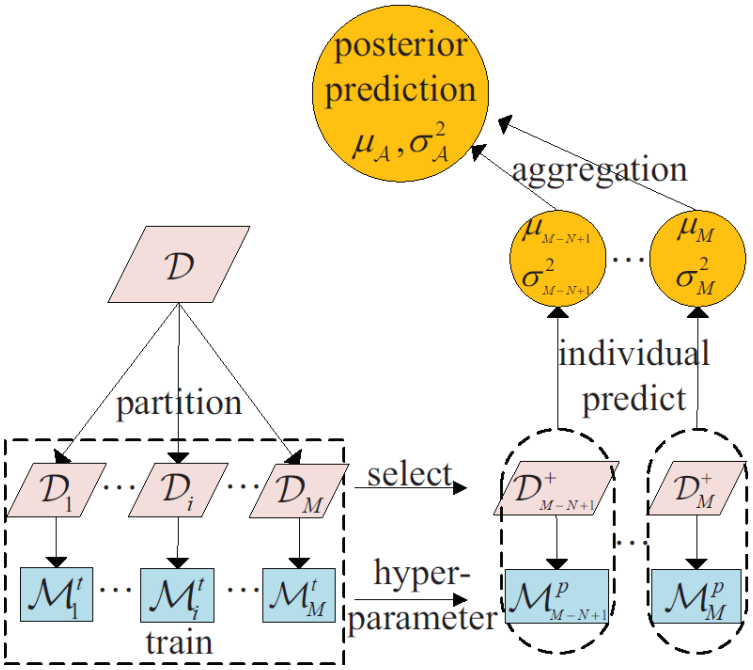

图 4.1 聚合模型的训练与预测过程

聚合模型遵循“分而治之”的思想，其实现过程由图 4.1 体现。首先，将整个数据集 $\mathcal{D}$ 划分为 M 个数据子集 $\{\mathcal{D}_i = \{\boldsymbol{X}_i, \boldsymbol{y}_i\}\}_{i=1}^M$，其中划分的方法可以通过随机抽样或者聚类算法[22]。其次，对于每一个数据子集 $\mathcal{D}_i$ 训练一个完整的高斯过程回归模型，一般被称为“专家”（experts）或“子模型”，记为 $\mathcal{M}_i^t$。假设每一个用于训练的子模型相互独立，则可以得到近似的边际似然函数用于训练，该函数有着如下的分解形式

$$p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\theta}) \approx \prod_{i=1}^{M} p(\boldsymbol{y}_i \mid \boldsymbol{X}_i, \boldsymbol{\theta}_i) \tag{4.1}$$

其中 $p(\boldsymbol{y}_i \mid \boldsymbol{X}_i, \boldsymbol{\theta}_i) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{K}_i + \sigma^2_{\varepsilon,i}\boldsymbol{I}_i)$，$\boldsymbol{K}_i = k(\boldsymbol{X}_i, \boldsymbol{X}_i) \in \mathbb{R}^{n_i \times n_i}$，$n_i$ 为第 i 个用于训练的子模型 $\mathcal{M}_i^t$ 的训练子集大小，通常我们有 $\sum_{i=1}^{M} n_i = n$。一般为了描述隐函数的全局特征，避免过拟合，如 Deisenroth 和 Ng（2015）[118]、Liu 等（2018）[120]建议在训练的时候所有子模型分享共同的超参数，有 $\boldsymbol{\theta}_i = \boldsymbol{\theta}_j$，其中 $1 \le i \ne j \le M$。明显的，超参数的减少同样有利于模型的训练。此外，为了保持模型的简洁，本文简单地为每一个子模型都分配大小约为 $n/M \ll n$ 的数据子集。

分享超参数的另一个好处为：预测时，可以调整每个子模型所拥有的数据子

[22]本文中聚类算法选用经典的 K-means 算法。

集。为了加以区分，新生成的 N 个用于预测的子模型记为$\left\{\mathcal{M}_i^p\right\}_{i=M-N+1}^{M}$，其中每个$\mathcal{M}_i^p$的超参数$\boldsymbol{\theta}$直接继承了用于训练的子模型$\mathcal{M}_i^t$训练所得的结果，而新的数据子集$\left\{\mathcal{D}_i^+\right\}_{i=M-N+1}^{M}$则由原数据子集$\left\{\mathcal{D}_i\right\}_{i=1}^{M}$组合而来。例如令$N=M$，则数据子集没有变化，或者令$\mathcal{D}_i^+=\mathcal{D}_i\cup\mathcal{D}_j$，其中$1\le i\ne j\le M$，此时用于预测的子模型$\mathcal{M}_i^p$则拥有着比训练的子模型$\mathcal{M}_i^t$更丰富的信息。给定$\mathcal{D}_i^+$，$\mathcal{M}_i^p$给出单个预测点$\boldsymbol{x}_*$上的后验分布$p\left(y_*\mid\mathcal{D}_i^+,\boldsymbol{x}_*\right)=\mathcal{N}\left(\mu_i\left(\boldsymbol{x}_*\right),\sigma_i^2\left(\boldsymbol{x}_*\right)\right)$，有

$$\mu_i\left(\boldsymbol{x}_*\right)=\boldsymbol{k}_{i*}^{\top}\left(\boldsymbol{K}_i+\sigma_\varepsilon^2\boldsymbol{I}_i\right)^{-1}\boldsymbol{y}_i, \tag{4.2}$$

$$\sigma_i^2\left(\boldsymbol{x}_*\right)=k_{**}-\boldsymbol{k}_{i*}^{\top}\left(\boldsymbol{K}_i+\sigma_\varepsilon^2\boldsymbol{I}_i\right)^{-1}\boldsymbol{k}_{i*}+\sigma_\varepsilon^2 \tag{4.3}$$

其中$\boldsymbol{k}_{i*}=k\left(\boldsymbol{X}_i,\boldsymbol{x}_*\right)$，这里及之后的$\boldsymbol{X}_i$都指来自于$\mathcal{D}_i^+$的。具体地，一致性从模型中体现为$\lim_{n\to\infty}\mu_i\left(\boldsymbol{x}_*\right)=\mu_f\left(\boldsymbol{x}_*\right)$，$\lim_{n\to\infty}\sigma_i^2\left(\boldsymbol{x}_*\right)-\sigma_\varepsilon^2=0$和$\lim_{n\to\infty}\sigma_\varepsilon^2=\sigma_f^2$，其中$\mu_f\left(\boldsymbol{x}_*\right)$和$\sigma_f^2$分别为隐函数真实的相应值与观测的方差。此时对于每一个预测的子模型$\mathcal{M}_i^p$都给出了一个预测结果，聚合模型则需要将这些预测结果$\left\{\mu_i\left(\boldsymbol{x}_*\right),\sigma_i^2\left(\boldsymbol{x}_*\right)\right\}_{i=M-N+1}^{M}$按一定的规则重新组合起来，以得到最终的预测结果。

对于不同的聚合模型，组合的规则则是不同的，但它们都是通过限制贝叶斯法则为模型提供可解释性。例如在不考虑实际的解释性的情况下，(G)PoE 假设$\mathcal{D}_i\perp\mathcal{D}_j$，(R)BCM 有条件假设$\mathcal{D}_i\perp\mathcal{D}_j\mid y_*$，GRBCM 则有$\mathcal{D}_i\perp\mathcal{D}_j\mid y_*,\mathcal{D}_c$，其中$\mathcal{D}_c$是从原始数据集中随机抽样出的数据子集，可记为$\mathcal{D}_c=\mathcal{D}_1$。此时，在预测点$\boldsymbol{x}_*$上的最终预测为

$$p_{\mathcal{A}}\left(y_*\mid\mathcal{D},\boldsymbol{x}_*\right)=\frac{\prod_{i=M-N+1}^{M}p_i^{\beta_i}\left(y_*\mid\mathcal{D}_i,\mathcal{D}_c,\boldsymbol{x}_*\right)}{p_c^{-1+\sum_{i=M-N+1}^{M}\beta_i}\left(y_*\mid\mathcal{D}_c,\boldsymbol{x}_*\right)} \tag{4.4}$$

其中β_i为第 i 个子模型$\mathcal{M}_i^p$的权重或者说贡献，具体的形式在下一节讨论。(G)PoE 模型只需考虑上式中黑色的部分，其中权重可以设定，具体地，令$\beta_i=1$则有 PoE。特别地，当$M=1$时(G)PoE 等价于完整的 GP 模型。(R)BCM 则明确地纳入了预测点的先验信息$p\left(y_*\mid\boldsymbol{x}_*\right)$，可由上式中黑色与红色的部分观测到。与(G)PoE 相似地是，(R)BCM 同样使用了事先设定的权重，且当$\beta_i=1$时可以得到 BCM。特别地，给定$\sum_{i=1}^{M}\beta_i=1$或$M=1$，RBCM 的预测则等同于 GPoE 的预测。然后，将蓝色部分的内容加入公式，则可得到 GRBCM 的预测公式。比较而言，(G)PoE 和(R)BCM 预测时运用了训练时的数据子集，有$\mathcal{D}_i^+=\mathcal{D}_i$，$N=M$，而 GRBCM 的预测则使用了增广的数据子集，有$\mathcal{D}_i^+=\mathcal{D}_c\cup\mathcal{D}_{i+1}$，$N=M-1$，$1\le i\le N$。此外，可以通过不同的假设看出，(G)PoE 通过对角阵近似完整的核矩阵，而(R)BCM 则在此基础上加入了先验信息，GRBCM 则使用了爪形矩阵。进一步，对于所有聚合模型，拓展假设则都可以使用信息更多的数据子集$\mathcal{D}_i^+=\mathcal{D}_c\cup\mathcal{D}_{i+1}$或$\mathcal{D}_i^+=\mathcal{D}_c\cup\mathcal{D}_{i+1}\cup\mathcal{D}_{i+2}$等。

不考虑 $\mathcal{D}_i^+$ 具体的组合方式，聚合模型的预测均值与预测方差可以写成如下形式：

$$\mu_{\mathcal{A}}(\boldsymbol{x}_*)=\sigma_{\mathcal{A}}^2(\boldsymbol{x}_*)\left[\sum_{i=M-N+1}^{M}\beta_i\sigma_i^{-2}(\boldsymbol{x}_*)\mu_i(\boldsymbol{x}_*)+\left(1-\sum_{i=M-N+1}^{M}\beta_i\right)\sigma_c^{-2}(\boldsymbol{x}_*)\mu_c(\boldsymbol{x}_*)\right] \tag{4.5}$$

$$\sigma_{\mathcal{A}}^{-2}(\boldsymbol{x}_*)=\sum_{i=M-N+1}^{M}\beta_i\sigma_i^{-2}(\boldsymbol{x}_*)+\left(1-\sum_{i=M-N+1}^{M}\beta_i\right)\sigma_{\mathcal{A}*}^{-2}(\boldsymbol{x}_*) \tag{4.6}$$

其中子模型 $\mathcal{M}_c^p$ 提供了全局预测 $p(y_*\mid\mathcal{D}_c,\boldsymbol{x}_*)=\mathcal{N}\left(\mu_c(\boldsymbol{x}_*),\sigma_c^2(\boldsymbol{x}_*)\right)$，(R)BCM 模型中 $\sigma_{\mathcal{A}*}^2(\boldsymbol{x}_*)=\sigma_{**}^2(\boldsymbol{x}_*)=k(\boldsymbol{x}_*,\boldsymbol{x}_*)+\sigma_\varepsilon^2$ 是先验的预测方差，而在 GRBCM 中则使用了信息更丰富的全局预测方差代替了先验方差，有 $\sigma_{\mathcal{A}*}^2(\boldsymbol{x}_*)=\sigma_c^2$。此时，除了不同模型中聚合方式的不同，还有权重的选择方式、数据子集的选取方式没有确定。因此，下一节则综合考虑权重、数据分块与聚合方式导致 GP 聚合模型不同的一致性。

4.2 收敛性

对于一些具体的 GP 聚合模型，在预测时不加上观测误差 σ_ε^2 的情况下，预测的非一致性已经被 Deisenroth 和 Ng（2015）[118]观察到，理论上的证明则由 Rullière 等（2018）[122]给出。并且，预测含观测误差的 GP 聚合模型的（非）一致性已经由 Liu 等（2018）[120]给出证明，其中考虑了数据分块使用了随机抽样与聚类算法的两类聚合模型。特别地，本文总结了：一预测点 $\boldsymbol{x}_*$ 远离输入数据 $\boldsymbol{X}$ 的时候，二预测点 $\boldsymbol{x}_*$ 落入输入数据 $\boldsymbol{X}$ 的时候，GP 聚合模型的一致性。并且，由此讨论了两者数据分区（随机分区和聚类分区）的优势，以及提供一致预测的 GPoE 和 GRBCM 的局限性。为了加以区分不同的数据分区方法，令输入数据 $\boldsymbol{X}$ 在输入域 $\Omega\in[0,1]^d$ 上是稠密的，即对于输入域上的任意一点 $\boldsymbol{x}'\in\Omega$，当 $n\to\infty$ 时有最小距离 $\lim_{n\to\infty}\min_{1\le i\le n}\|\boldsymbol{x}'-\boldsymbol{x}_i\|=0$。

首先讨论预测点 $\boldsymbol{x}_*$ 远离输入数据 $\boldsymbol{X}$ 的情况。给定预测点 $\boldsymbol{x}_*$ 的相对距离 $r_i=\min\|\boldsymbol{x}_*-\boldsymbol{x}\|_{\forall\boldsymbol{x}\in\boldsymbol{X}_i}$，每个子模型相当于完整的 GP 模型给出预测方差 $\lim_{r_i\to\infty}\sigma_i^2(\boldsymbol{x}_*)=\sigma_{**}^2(\boldsymbol{x}_*)$，这同样说明，理想的 GP 聚合模型的预测方差也因收敛到先验 $\sigma_{**}^2(\boldsymbol{x}_*)$。由预测公式（4.5）、（4.6）可得，PoE 产生了“过于自信”的预测，因为 $\sigma_{PoE}^2(\boldsymbol{x}_*)\to_{r\to\infty}1/M\sigma_{**}^2(\boldsymbol{x}_*)$。相反，传统的 GPoE（tGPoE）使用了变化的权重（即对于不同的子模型，权重不同）$\beta_i=0.5\left(\log\sigma_{**}^2(\boldsymbol{x}_*)-\log\sigma_i^2(\boldsymbol{x}_*)\right)$，该权重表示了先验 $p(y_*\mid\boldsymbol{x}_*)$ 与后验 $p(y_*\mid\mathcal{D}_i,\boldsymbol{x}_*)$ 在微分熵上的差异，而这权重在 $\boldsymbol{x}_*$ 远离 $\boldsymbol{X}$ 时会收敛到 0，导致了“爆炸”的预测方差，即 $\sigma_{tGPoE}^2(\boldsymbol{x}_*)\to_{r\to\infty}\infty$。为了解决该问题，其中一种方法是为 GPoE 使用更简单的权重 $\beta_i=1/M$ [116]，这可以直接导致理想的预测方差 $\sigma_{GPoE}^2(\boldsymbol{x}_*)\to_{r\to\infty}\sigma_{**}^2(\boldsymbol{x}_*)$。另一种方法则是使用(R)BCM 模

型，该类模型使用的先验信息可直接获得理想的预测方差 $\sigma^2_{(R)BCM}(\boldsymbol{x}_*)\to_{r\to\infty}\sigma^2_{**}(\boldsymbol{x}_*)$。特别地，RBCM 还可以保留这种变化的权重，其中名字中的 R 是 robust 的缩写，因为该变化的权重可以降低预测能力弱的子模型，从而使模型的预测更加鲁棒。对于 GRBCM，根据权重

$$\beta_i=\begin{cases}1, i=2,\\ 0.5\left(\log\sigma_c^2(\boldsymbol{x}_*)-\log\sigma_i^2(\boldsymbol{x}_*)\right), 3\le i\le M\end{cases} \tag{4.7}$$

可得 $\sigma^2_{GRBCM}(\boldsymbol{x}_*)\to_{r\to\infty}\sigma^2_{**}(\boldsymbol{x}_*)$。当 $i=2$ 时，权重 β_i 保持着常数，保证 GRBCM 的预测方差收敛至 $\sigma^2_{**}(\boldsymbol{x}_*)$，而当 $3\le i\le M$ 时，β_i 则以变化的权重体现作用，随着数据量大小 n 的增加该权重将收敛至 0。

表 4.1　预测点 $\boldsymbol{x}_*$ 落入输入数据 $\boldsymbol{X}$ 时，使用随机分区的预测均值与方差的收敛性，其中 $a=\sigma_f^{-2}\big/\left(\sigma_f^{-2}-\sigma_{**}^{-2}(\boldsymbol{x}_*)\right)\ge 1$。

	PoE	tGPoE	GPoE	BCM	RBCM	GRBCM
$\mu_{\mathcal{A}}(\boldsymbol{x}_*)\to$	$\mu_f(\boldsymbol{x}_*)$	$\mu_f(\boldsymbol{x}_*)$	$\mu_f(\boldsymbol{x}_*)$	$a\mu_f(\boldsymbol{x}_*)$	$a\mu_f(\boldsymbol{x}_*)$	$\mu_f(\boldsymbol{x}_*)$
$\sigma^2_{\mathcal{A}}(\boldsymbol{x}_*)\to$	0	0	σ_f^2	0	0	σ_f^2

然后，本文讨论预测点 $\boldsymbol{x}_*$ 落入输入数据 $\boldsymbol{X}$ 的情况。特别地，该情况可以按数据分区的方式划分为两种子情况，即随机分区和聚类分区。给定一些条件，并令子集大小满足 $n/M\to_{n\to\infty}\infty$，当数据子集 $\boldsymbol{X}_i$ 是从数据 $\boldsymbol{X}$ 中的不放回抽样取得的，则有一致性[23] $\lim_{n\to\infty}\mu_i(\boldsymbol{x}_*)=\mu_f(\boldsymbol{x}_*)$ 和 $\lim_{n\to\infty}\sigma_i^2(\boldsymbol{x}_*)=\sigma_f^2$。然后，根据预测公式（4.5）、（4.6），使用随机分区的 GP 聚合模型的预测一致性列在表 4.1 中。对于 PoE 类，预测方差有 $\lim_{n\to\infty}\sigma_{\mathcal{A}}^{-2}(\boldsymbol{x}_*)=M\bar{\beta}\sigma_f^{-2}$，其中 $\bar{\beta}=\lim_{n\to\infty}\beta_i$。PoE 与 tGPoE 分别使用了权重 $\bar{\beta}=1$ 和 $\bar{\beta}=0.5\log\left(\sigma^2_{**}(\boldsymbol{x}_*)\big/\sigma_f^2\right)$，两者都因为 $\lim_{n\to\infty}M=\infty$ 提供了不一致的预测，但使用 $\bar{\beta}=1/M$ 的 GPoE 则不存在该问题。幸运的是，PoE 类的预测不一致性只体现在预测的方差上，而预测的均值是一致的，有 $\lim_{n\to\infty}\mu_{\mathcal{A}}(\boldsymbol{x}_*)=M\bar{\beta}\sigma_f^{-2}\mu_f(\boldsymbol{x}_*)\big/M\bar{\beta}\sigma_f^{-2}=\mu_f(\boldsymbol{x}_*)$。而对于当前情况下的(R)BCM，同样存在着预测不一致的问题，其中预测方差有 $\lim_{n\to\infty}\sigma_{\mathcal{A}}^{-2}(\boldsymbol{x}_*)=M\bar{\beta}\left(\sigma_f^{-2}-\sigma_{**}^{-2}(\boldsymbol{x}_*)\right)+\sigma_{**}^{-2}(\boldsymbol{x}_*)\to_{M\to\infty}\infty$，不过可以发现，令权重为 $\beta_i=1/M$ 时该预测方差可以收敛到真实的预测方差 σ_f^2，而当 $M\to\infty$ 时预测均值有 $\lim_{n\to\infty}\mu_{\mathcal{A}}(\boldsymbol{x}_*)=M\bar{\beta}\sigma_f^{-2}\mu_f(\boldsymbol{x}_*)\big/M\bar{\beta}\left(\sigma_f^{-2}-\sigma_{**}^{-2}(\boldsymbol{x}_*)\right)+\sigma_{**}^{-2}(\boldsymbol{x}_*)\ge\mu_f(\boldsymbol{x}_*)$，显然当 $\sigma_{**}^{-2}(\boldsymbol{x}_*)\to 0$ 时预测的均值才一致。对于 GRBCM，预测公式（4.5）、（4.6）可以重新写成

$$\mu_{GRBCM}(\boldsymbol{x}_*)=\sigma^2_{GRBCM}(\boldsymbol{x}_*)\left[\sigma_2^{-2}(\boldsymbol{x}_*)\mu_2(\boldsymbol{x}_*)+\sum_{i=3}^{M}\beta_i\left(\sigma_i^{-2}(\boldsymbol{x}_*)\mu_i(\boldsymbol{x}_*)-\sigma_c^{-2}(\boldsymbol{x}_*)\mu_c(\boldsymbol{x}_*)\right)\right] \tag{4.8}$$

[23]注：关于样本内估计的一致性理论可见 Choi 和 Schervish（2004）[21]，关于样本外预测的一致理论可见 Vazquez 和 Bect（2009）[20]。

$$\sigma_{GRBCM}^{-2}\left(\boldsymbol{x}_*\right)=\sigma_2^{-2}\left(\boldsymbol{x}_*\right)+\sum_{i=3}^{M}\beta_i\left(\sigma_i^{-2}\left(\boldsymbol{x}_*\right)-\sigma_c^{-2}\left(\boldsymbol{x}_*\right)\right) \tag{4.9}$$

其中全局子模型提供了一致性预测（即 $\mu_2\left(\boldsymbol{x}_*\right)\to_{n\to\infty}\mu_f\left(\boldsymbol{x}_*\right)$，$\sigma_2^2\left(\boldsymbol{x}_*\right)\to_{n\to\infty}\sigma_f^2$），而其他子模型的作用则是在此基础上修正，并且当 $M\to\infty$ 时几乎没有影响，即 $\beta_i\left(\sigma_i^{-2}\left(\boldsymbol{x}_*\right)\mu_i\left(\boldsymbol{x}_*\right)-\sigma_c^{-2}\left(\boldsymbol{x}_*\right)\mu_c\left(\boldsymbol{x}_*\right)\right)\to_{n\to\infty}0$ 和 $\beta_i\left(\sigma_i^{-2}\left(\boldsymbol{x}_*\right)-\sigma_c^{-2}\left(\boldsymbol{x}_*\right)\right)\to_{n\to\infty}0$，故导致了 GRBCM 的一致预测。但是，当 $M\to_{n\to\infty}\infty$ 时，GRBCM 需要一个更大的子集大小 n/M 来维持预测的一致性，如满足 $\lim_{n\to\infty}n^2/M>0$，细节请见 Liu 等（2018）[120]。

表 4.2 预测点 $\boldsymbol{x}_*$ 落入输入数据 $\boldsymbol{X}$ 时，使用随机分区的预测均值与方差的收敛性，其中 $a=\sigma_f^{-2}\Big/\left(\sigma_f^{-2}-\sigma_{**}^{-2}\left(\boldsymbol{x}_*\right)\right)\geq 1$。

	PoE	tGPoE	GPoE	BCM	RBCM	GRBCM
$\mu_{\mathcal{A}}\left(\boldsymbol{x}_*\right)\to$	$\mu_f\left(\boldsymbol{x}_*\right)$	$\mu_f\left(\boldsymbol{x}_*\right)$	×	$a\mu_f\left(\boldsymbol{x}_*\right)$	$a\mu_f\left(\boldsymbol{x}_*\right)$	$\mu_f\left(\boldsymbol{x}_*\right)$
$\sigma_{\mathcal{A}}^2\left(\boldsymbol{x}_*\right)\to$	0	0	$\sigma_f^2<\sigma_{GPoE}^2\left(\boldsymbol{x}_*\right)<\sigma_{**}^2\left(\boldsymbol{x}_*\right)$	0	0	σ_f^2

对于聚类分区，则可以考虑两个有代表性的数据子集，即离预测点 $\boldsymbol{x}_*$ 最近的子集 $\boldsymbol{X}_a$ 和最远的子集 $\boldsymbol{X}_b$。满足一定条件时，使用子集 $\boldsymbol{X}_a$ 的子模型可以提供一致的预测，有 $\lim_{n\to\infty}\mu_a\left(\boldsymbol{x}_*\right)=\mu_f\left(\boldsymbol{x}_*\right)$ 和 $\lim_{n\to\infty}\sigma_a^2\left(\boldsymbol{x}_*\right)=\sigma_f^2$ [20]。因为输入域是有界的，所以可得 $\lim_{n\to\infty}\sigma_b^2\left(\boldsymbol{x}_*\right)<\sigma_{**}^2\left(\boldsymbol{x}_*\right)$。因此有 $\sigma_f^2\leq\lim_{n\to\infty}\sigma_i^2\left(\boldsymbol{x}_*\right)<\sigma_{**}^2\left(\boldsymbol{x}_*\right)$，此时预测的一致性可见表 4.2，这些结果同样可以通过公式（4.5）、（4.6）得到。给定平稳过程，因为子模型给出的估计为无偏估计，有 $\mathbb{E}\mu_i\left(\boldsymbol{x}_*\right)=\mu_f\left(\boldsymbol{x}_*\right)$。虽然 (tG)PoE 和(R)BCM 给出了“过于自信”的预测方差，但它们仍能保证预测的均值是收敛的。对于 GPoE 而言，有 $\lim_{n\to\infty}\sigma_\varepsilon^2\left(\boldsymbol{x}_*\right)=\sigma_f^2$ 且 $\sigma_{\mathcal{A}}^2\left(\boldsymbol{x}_*\right)-\sigma_f^2>0$ 暗示着其预测均值并不能收敛，即使是 $n\to\infty$ 的情况。此时，只有 GRBCM 给出了一致的预测，其得益于随机分区而得的全局数据子集 $\mathcal{D}_c$。

进一步，本文将讨论一下不同分区方式的优势与劣势。之前对于不同分区，GP 聚合模型的一致性体现了那最接近 $\boldsymbol{x}_*$ 的一个数据子集是一致性的来源。首先，给定 $n/M\to\infty$，M 足够大但是是有限的，此时则有 $\sigma_{i,r}^2\left(\boldsymbol{x}_*\right)<\sigma_{i,k}^2\left(\boldsymbol{x}_*\right)$，其中 $1\leq i\leq M$，r 表示随机分区，k 表示聚类分区，则除了 GRBCM 可以得出结论 $\sigma_{\mathcal{A},r}^2\left(\boldsymbol{x}_*\right)<\sigma_{\mathcal{A},k}^2\left(\boldsymbol{x}_*\right)$。对于(R)BCM 而言，预测的均值则有 $\mu_f\left(\boldsymbol{x}_*\right)\leq\mu_{\mathcal{A},r}\left(\boldsymbol{x}_*\right)<\mu_{\mathcal{A},k}\left(\boldsymbol{x}_*\right)$。由此可见，当数据子集大小足够大时，随机分区会比聚类分区更快得收敛。对于 GRBCM 而言，$\beta_i\left(\sigma_i^{-2}\left(\boldsymbol{x}_*\right)-\sigma_c^{-2}\left(\boldsymbol{x}_*\right)\right)\to 0$，$2<i\leq M$，意味着不同的分区方式在理论上不影响聚合模型的收敛性。然后，考虑含有更少数据点的子模型，即 $M\to_{n\to\infty}\infty$。这种情况则可以直接由表 1 和表 2 看到，随机分区仍然在理论方面具有一定的优势。然而，在现实应用当中，子模型等同于完整的 GP 模型，其数据子集大小 n/M 仍被限制在了 $\mathcal{O}\left(10^4\right)$。在这种

情况下，使用随机分区的子模型则可能由于不充分的信息而使得其难以建模快速变化的特征[24]，因为当数据点不充分时，根据极大似然估计得出的 GP 模型会倾向于平滑掉一些快速的波动。相比之下，基于聚类分区的子模型则在局部给出了更有效的预测，因为其在局部具有更高的数据密度，并且聚类分区在中小数据集实证方面的优势可以参见 Liu 等（2018）[120]、Rullière 等（2018）[122]。为了优先保证 GP 聚合模型的预测是一致的，在没有特别说明的情况下，本文使用随机分区划分数据子集。特别地，在给出说明的情况下，本文使用下标“r”代表随机分区，用下标“k”代表使用 K-means 算法的聚类分区。

最后，本小节讨论一致的 GPoE 和 GRBCM 的特点。明显地，GPoE 非常适合处理数据量巨大（如 $n>10^6$）、特征变化不快的数据集，因为使用了随机分区。相反，虽然 GRBCM 被数据子集的大小所限制，但是当数据集的大小相对较少时（如 $10^4 \le n \le 10^5$），GRBCM 的预测效果更佳，因为在预测时子模型使用了信息更丰富的增广数据子集 $\mathcal{D}_i^+$，它在全局子集 $\mathcal{D}_c$ 上进行了修正。鉴于 GPoE 和 GRBCM 在不同数据集上的有效性，则可以拓展两者模型使得其对于多种大小的数据集依然保持有效的一致性，尤其是对于大数据集。

4.3 双层聚合模型

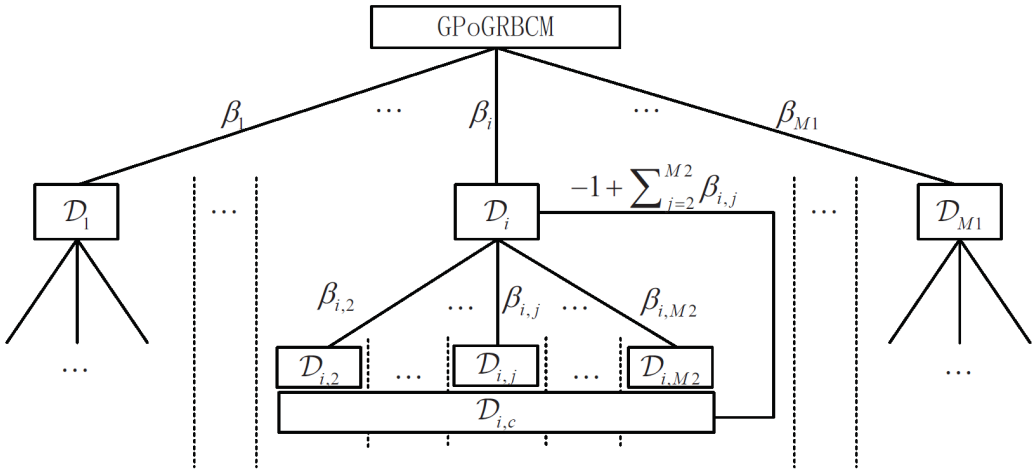

图 4.2　双层聚合模型 GPoGRBCM 结构

本文提出的 GPoGRBCM 是一个嵌套的二层模型，即将 GRBCM 嵌入 GPoE 以替代其原本的完整 GP 模型，这样可以近似地拓宽子模型的限制，模拟子模型在更多数据点上预测的结果。GPoGRBCM 的结构如图 4.2 所示。在 GP 聚合模型框架下，首先使用随机抽样将数据集 $\mathcal{D}$ 划分为 M1 个数据子集 $\{D_i\}_{i=1}^{M1}$，使得所有的第一层数据子集都能够描绘隐函数的全局特征。然后，将 GRBCM 替代原始的完整 GP，进一步将 $\mathcal{D}_i$ 划分为更小的数据子集 $\{\mathcal{D}_{i,j}\}_{i=1,j=1}^{i=M1,j=M2}$ 用于训练，并且组

[24]快速变化的特征指的是在单位输入域范围内，真实的输出随着输入变化的变化范围非常大。

成一系列的增广数据集$\left\{\mathcal{D}_{i,j}^{+}=\left\{\mathcal{D}_{i,c},\mathcal{D}_{i,j}\right\}\right\}_{j=2}^{M2}$用于预测。给定独立性假设$\mathcal{D}_i \perp \mathcal{D}_k$对于$1\le i\ne k\le M1$和$\mathcal{D}_{i,j}\perp\mathcal{D}_{i,l}\,|\,\mathcal{D}_{i,c},y_*$对于$2\le j\ne l\le M2$，并且权重也可以由 GPoE 和 GRBCM 中得到

$$\begin{cases}\beta_i=1/M1, 1\le i\le M1,\\ \beta_{i,j}=1, j=2,\\ \beta_{i,j}=0.5\left(\log\sigma_{i,c}^2\left(\boldsymbol{x}_*\right)-\log\sigma_{i,j}^2\left(\boldsymbol{x}_*\right)\right), 3\le j\le M2,\end{cases} \tag{4.10}$$

可以通过贝叶斯法则按如下方式近似精确的后验分布$p\left(y_*\,|\,\mathcal{D},\boldsymbol{x}_*\right)$：

$$p\left(y_*\,|\,\mathcal{D},\boldsymbol{x}_*\right)\approx\prod_{i=1}^{M1}p^{\beta_i}\left(y_*\,|\,\mathcal{D}_i,\boldsymbol{x}_*\right)=\prod_{i=1}^{M1}\frac{\prod_{j=2}^{M2}p_{i,j}^{\beta_i\beta_{i,j}}\left(y_*\,|\,\mathcal{D}_{i,j},\mathcal{D}_{i,c},\boldsymbol{x}_*\right)}{p_{i,c}^{\beta_i\sum_{j=2}^{M2}\beta_{i,j}}\left(y_*\,|\,\mathcal{D}_{i,c},\boldsymbol{x}_*\right)} \tag{4.11}$$

而预测的均值与方差可以如下表示

$$\mu_{GPoGRBCM}\left(\boldsymbol{x}_*\right)=\sigma_{GPoGRBCM}^2\left(\boldsymbol{x}_*\right)\sum_{i=1}^{M1}\beta_i\sigma_i^2\left(\boldsymbol{x}_*\right)\mu_i\left(\boldsymbol{x}_*\right), \tag{4.12}$$

$$\sigma_{GPoGRBCM}^{-2}\left(\boldsymbol{x}_*\right)=\sum_{i=1}^{M1}\beta_i\sigma_i^{-2}\left(\boldsymbol{x}_*\right) \tag{4.13}$$

其中$\mu_i\left(\boldsymbol{x}_*\right)$和$\sigma_i^2\left(\boldsymbol{x}_*\right)$在 GPoE 中为完整的 GP 给出的预测结果，而在此处则由 GRBCM 替代：

$$\mu_i\left(\boldsymbol{x}_*\right)=\sigma_i^2\left(\boldsymbol{x}_*\right)\left[\sum_{j=2}^{M2}\beta_{i,j}\sigma_{i,j}^{-2}\left(\boldsymbol{x}_*\right)\mu_{i,j}\left(\boldsymbol{x}_*\right)-\left(\sum_{j=2}^{M2}\beta_{i,j}-1\right)\sigma_{i,c}^{-2}\left(\boldsymbol{x}_*\right)\mu_{i,c}\left(\boldsymbol{x}_*\right)\right], \tag{4.14}$$

$$\sigma_i^{-2}\left(\boldsymbol{x}_*\right)=\sum_{j=2}^{M2}\beta_{i,j}\sigma_{i,j}^{-2}\left(\boldsymbol{x}_*\right)-\left(\sum_{j=2}^{M2}\beta_{i,j}-1\right)\sigma_{i,c}^{-2}\left(\boldsymbol{x}_*\right) \tag{4.15}$$

从该模型的结构中可以清楚地发现，当$M1=1$时 GPoGRBCM 退化为 GRBCM，并且当$M2=1$时 GPoGRBCM 退化为 GPoE。对于预测的极限性质，当预测点$\boldsymbol{x}_*$远离输入数据$\boldsymbol{X}$时，GPoGRBCM 的预测能够返回到先验分布，并且，给定$n/M1/M2\to_{n\to\infty}\infty$，$\lim_{n\to\infty}n/M1/M2^2>0$[25]和零均值的平稳核函数，当预测点落入输入域$\Omega$时有

$$\lim_{n\to\infty}\mu_{GPoGRBCM}\left(\boldsymbol{x}_*\right)=\mu_f\left(\boldsymbol{x}_*\right) \tag{4.16}$$

$$\lim_{n\to\infty}\sigma_{GPoGRBCM}^2\left(\boldsymbol{x}_*\right)=\sigma_f^2 \tag{4.17}$$

若$M1\times M2=M$，GPoGRBCM 则在时间复杂度上与(G)PoE 和(R)BCM 处于同一水平。但是，事实上 GPoGRBCM 的运算速度总会比上述模型大一些，因为当数据子集分好块以后，该模型需要像 GRBCM 一样转置一个更大的增广矩阵。假如每一个数据子集大小同为$n/M1/M2$，则 GPoGRBCM 预测的时间复杂度为

[25] 给定$M=M1\times M2$，GRBCM 需要相对更大的数据子集来确保$\lim_{n\to\infty}n/M1^2/M2^2>0$。

$O\left(n^2/M1/M2\right)$，主要花费在聚合子模型给出的后验方差上。而 GPoGRBCM 的训练时间与其他的聚合 GP 一样，因为训练的过程都是通过类似 PoE 的聚合方式，但其可以有效地减少数据集在分区方面所耗费的时间，如：当数据量 $n>10^5$ 时，原始的 K-means 算法在聚类时会花费大量的时间且难以收敛。

最后，这个双层的聚合模型同样如分布式 GP 一样，可以拓展到更深的多层模型，而 GPoGRBCM 的一致性也以为着可以任意组合 GPoE 和 GRBCM 而不失一致性。但值得注意的是，多层的 GPoE 与单层的 GPoE 在设置相同时返回的预测结果是相同的[118]，其仅仅拓展了模型的计算性。因此，GRBCM 可以为其上层的子模型提供更精确的预测，因为其本身在预测时，除了全局的子模型，其他子模型的作用是修正全局子模型给出的预测。总而言之，聚合 GP 的多层模型确实能够通过分布式以及并行运算来降低计算所消耗的时间。

4.4 数值实验

本文通过两种指标评估预测结果，因为 GP 模型的优势在于其不仅能给出预测，而且可以度量预测的不确定性，故一类指标衡量预测的精确性，而第二类指标则主要描述预测分布的准确性。首先，本文使用标准化的均方误差（standardized mean square error，SMSE）来度量模型预测的精确性，其公式可以定义如下

$$SMSE=\frac{\sum_{k=1}^{n_*}\left(y_{*k}-\mu_{*k}\right)^2}{\sum_{k=1}^{n_*}\left(y_{*k}-\bar{\boldsymbol{y}}\right)^2} \tag{4.18}$$

其中 y_{*k} 和 μ_{*k} 分别表示第 k 个预测点 $\boldsymbol{x}_*$ 上的真实目标值与模型预测的均值。SMSE 可以看作衡量预测均值与真实隐函数的值之间差异的标准，并且当我们使用训练集中的全体输出 $\boldsymbol{y}$ 的均值作为预测时，其值为 1。此外，为了度量预测的后验分布的精确性，本文使用了平均标准化对数损失（mean standardized log loss，MSLL）

$$MSLL=\frac{1}{n_*}\sum_{k=1}^{n_*}\left[\log\mathcal{N}\left(\bar{\boldsymbol{y}},\mathrm{var}\left(\boldsymbol{y}\right)\right)-\log p\left(y_{*k}\mid\mathcal{D},\boldsymbol{x}_{*k}\right)\right] \tag{4.19}$$

其中 $\log p\left(y_{*k}\mid\mathcal{D},\boldsymbol{x}_{*k}\right)=-0.5\left[\log\left(2\pi\sigma_{*k}^2\right)+\left(y_{*k}-\mu_{*k}\right)^2/\sigma_{*k}^2\right]$。具体地，MSLL 越小表示模型的质量越高，而当模型给出与全体训练输出 $\boldsymbol{y}$ 的均值与方差时，MSLL 将会等于 0。

4.4.1 玩具数据集

该节内容主要通过一个简单函数生成的数据集来研究前文提到过的各种聚合 GP 模型，该函数可以表示如下：

$$y\left(x\right)=5x^2\sin\left(12x\right)+\left(x^3-0.5\right)\sin\left(3x-0.5\right)+4\cos\left(2x\right)+\varepsilon \tag{4.20}$$

其中$\varepsilon \sim \mathcal{N}(0,0.25)$。依据 Liu 等（2018）[120]的设置，我们随机生成多个数据集，其中用于训练的数据集的大小分别为$n=10^4,5\times10^4,10^5,5\times10^5,10^6$，取值区间为$[0,1]$，而用于测试的数据集则随机从$[-0.2,1.2]$中生成$n_*=0.1n$个测试点。数据预处理时，本文先将$\boldsymbol{X}$的每个特征以及$\boldsymbol{y}$标准化至均值为 0 和方差为 1。每个子模型被分配的数据量大小为$n/M(n/M1/M2)=500$。本文通过 GPML v4.2 工具箱[150]和 Liu 等（2018）[120]分享的代码实现聚合 GP 模型，并且，核函数使用 SEARD 核函数，有初始化参数$\sigma_l^2=1$，$l_i=1$和$\sigma_\varepsilon^2=0.1$，而优化算法则使用的是共轭梯度下降法，本文将下降的最大步数设为了 25。所有的代码在一台普通的个人电脑上运行，其配置为 3.60GHz 和 8GB RAM。

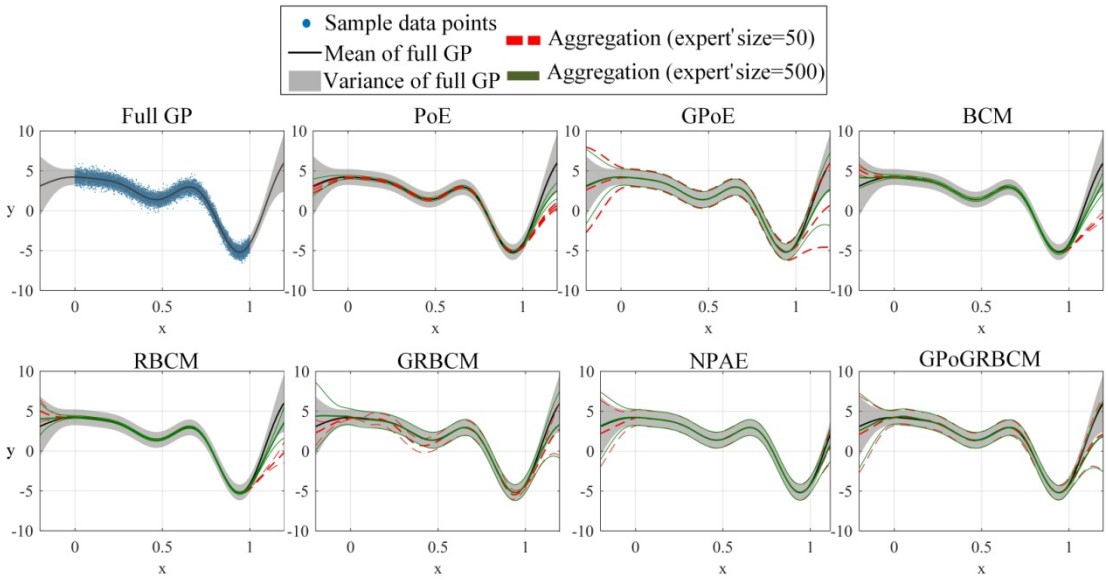

图 4.3　聚合 GP 模型应用于玩具数据集的结果，其中子模型的大小n/M（$n/M1/M2$）分别为 50 和 500，数据集的大小n为 10000，$n/M1$表示 GPoGRBCM 的第一层大小。

图 4.3 展现了 7 种聚合 GP 模型与完整 GP 模型在数据量为$n=10^4$下比较的结果，其中数据子集的大小展示了$n/M(n/M1/M2)=50$和 500 的两种情况。依据此图，可以直观地理解本文所说的预测的“有效”一致性。根据 2.3 节的极限理论，PoE 和(R)BCM 明显不能收敛至含噪声的隐函数。相反，其他的聚合 GP 模型则随着子数据集大小的提升而给出了一致的预测。然而当数据子集大小为$n/M=50$时，GRBCM 在$[0,1]$上给出了局部很奇怪的预测，这也暗示着太小的数据子集不能满足条件$\lim_{n\to\infty} n^2/M>0$，从而导致预测的不一致性。对于 GPoGRBCM，令第一层子集大小$n/M1=2000$，其得益于 GPoE 和 GRBCM 的特色，成功地产生了接近于 NPAE 的一致预测结果。此外，当数据子集所包含的数据量过少时，明显地能观察到该模型在区间$[1,1.2]$上给出了过于保守的预测方差。

图 4.4 则给出了聚合 GP 模型基于不同数据集大小 n 的比较结果。图 4(a)和(b)显示了三种模型所消耗的时间—数据分块时间、训练时间、预测时间。注意

NPAE 在处理 $n \geq 5\times10^4$ 的数据集时花费的时间会超过一星期，并且 GRBCM$_k$ 由聚类算法所消耗的时间在 $n > 5\times10^5$ 时爆炸增长。相较于 GRBCM$_k$,GPoGRBCM$_k$ 则通过分块以后再聚类的方法有效地解决了聚类算法耗时长的问题。此外，图 4(c)和(d)显示了除 GRBCM 外的聚合 GP 模型在所给的数据大小范围内保持着稳定的预测，其中 GPoGRBCM 给出了最优的预测。对于 GRBCM 而言，从预测的表现结果可以发现在 $n > 10^5$ 的情况下，其难以通过固定的数据子集大小 $n/M = 500$ 来保证预测的一致性。

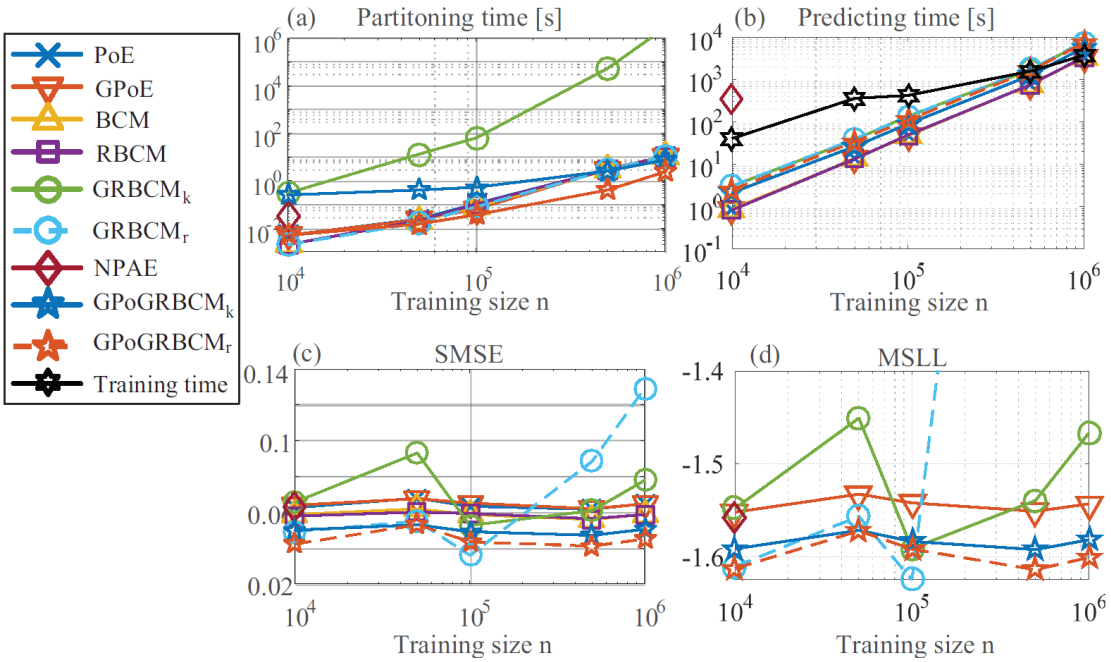

图 4.4 不同聚合 GP 模型在玩具数据集上关于运行时间、SMSE、MSLL 的比较

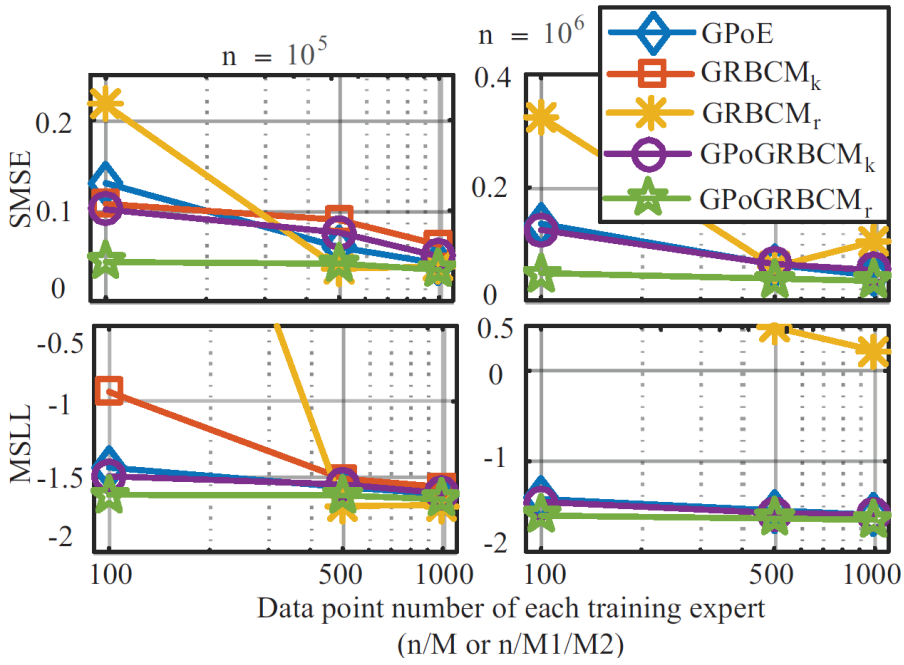

图 4.5 不同 GP 聚合模型在玩具数据集上随着子模型大小变化的一致性，其中数据集大小

n分别为10^5和10^6。

图4.5则比较了一致的聚合GP模型对于数据子集大小的接受能力。本文以数据集大小$n=10^5$和10^6为例，比较了当数据子集大小为$n/M\left(n/M1/M2\right)=100,500,1000$时提供一致预测的聚合GP模型的预测结果。GRBCM的结果显示了其需要更大的数据子集大小来满足一致预测的条件，如当$n=10^5$时有$n/M\geq 500$，或当$n=10^6$时有$n/M>1000$。GPoGRBCM则产生了对于数据子集大小而言更加稳健的预测结果，尤其是当数据子集很小时，其预测结果明显优于GPoE。

表 4.3　当数据集大小为10^6时，GPoGRBCM使用不同分区方法与不同的第一层$n/M1$、第二层$n/M1/M2$大小的结果比较。

GPoGRBCM$_k$的 SMSE(MSLL)				
n/M1/M2	n/M1			
	2000	5000	10000	100000
100	0.1029(-1.4940)	0.1249(−1.4525)	0.1183(−1.4715)	0.0562(−0.3044)
500	0.0684(−1.5579)	0.0645(−1.5737)	0.0718(−1.5538)	0.0936(−1.5062)
1000	**0.0348**(**−1.6457**)	0.0539(−1.5944)	0.0562(−1.5928)	0.0858(−1.5008)
GPoGRBCM$_r$的 SMSE(MSLL)				
n/M1/M2	n/M1			
	2000	5000	10000	100000
100	0.0688(−1.5541)	0.0486(−1.6019)	0.0460(−1.6057)	0.1422(6.1674)
500	0.0450(−1.6109)	0.0385(−1.6330)	0.0380(−1.6433)	0.0346(**−1.6812**)
1000	0.0351(−1.6443)	0.0350(−1.6479)	**0.0319**(−1.6621)	0.0495(−1.6117)

接着考虑GPoGRBCM的第一层子集大小$n/M1$，其实验结果如表4.3所示。注意，由理论可得，当第一层的子集大小非常小时，GPoGRBCM的预测会更接近于GPoE，而当第一层的子集大小非常大时则GPoGRBCM的预测会更接近于GRBCM。GPoGRBCM$_k$的最优预测是在第一层子集大小非常小时，这也意味着GRBCM过度的修正效应并没有起到好的作用。相反，GRBCM配合第二层数据子集是随机分区的时候产生了良好的预测，因为对于公式（4.20）这种变化缓慢的函数，使用随机分区去修正全局信息是一个比较稳健的保持一致性的方法。

4.4.2 现实数据集

本节当中，将比较各种聚合GP模型在一系列UCI数据集上的综合表现。除了song数据集，每一个数据集将随机地按4:1的比例分为训练集与测试集[26]。superconductor[151]数据集包含了一系列与超导体临界温度有关的特征。protein[152]数据集包含了蛋白质三级结构的理化性质。wec[153][154]数据集由波能转换器的位

[26] song数据集本身自带了训练集与测试集，为了避免专辑生产的时间等因素产生的影响。

置与吸收的波强组成。song[155]数据集则是通过音频特征推断音乐发行的年份。electric[152]数据集则是一系列关于家庭电能消耗的指标。co[156][157]数据集则用来估计 CO 的密度，根据温度调制的金属氧化物气体传感器获取的数据。本文通过上述 6 个数据集评估聚合 GP 模型的能力。其中数据集的一些参数由表 4.4 给出，数据子集个数 M 是事先设定的参数，同理(M1,M2)是模型 GPoGRBCM 中的，而设定时考虑了数据子集的大小，个数多少是为了确保每个数据子集拥有约 1000 个数据。特别地，对于数据集 co，本文保证了每个数据子集的大小在 500 左右，以节约模型在大数据上运行的时间。为了保证 GPoGRBCM 与其他模型是可比较的，本文令 $M1\times M2\approx M$ 。其他的一些设置如 4.4.1 节所示。

表 4.4　六个现实数据集的特征以及模型参数（子模型大小）设置

数据集	n（训练样本）	n_*（测试样本）	D（输入维）	M (M1,M2)
superconductor	17 011	4252	81	18 (3,6)
protein	36 584	9146	9	36 (6,6)
wec	230 400	57 599	48	230 (10,23)
song	463 715	51 630	90	460 (23,20)
electric	1 639 424	409 856	6	1639 (80,20)
co	3 074 528	768 632	18	6160 (77,80)

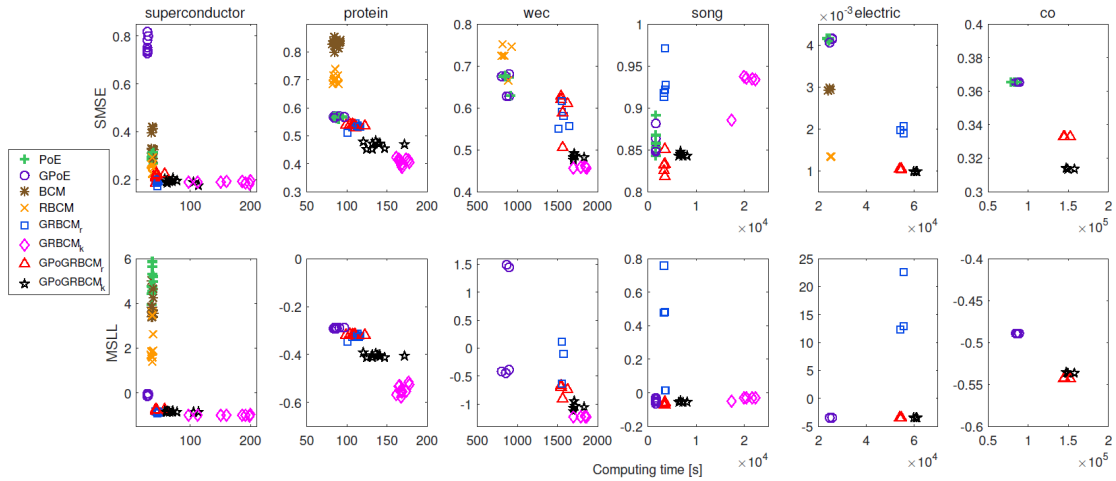

图 4.6　不同聚合 GP 模型在六个现实数据集上的结果。实验使用 Matlab 的四核并行运算。对于左边两个数据集，实验重复了 10 次；对于中间的两个数据集，实验重复了 5 次；对于右边的两个数据集，实验重复了 3 次。GRBCM$_k$ 因为在分区时花费了过量的时间，故在右边两个数据集上没有数据点。

纵观图 4.6，BCM 和 RBCM 总是给出了较差的预测均值与预测方差。PoE 与 GPoE 经常给出相似的预测均值，且预测表现在大数据集上表现较优。但是 PoE 总是给出了较差的预测方差，这点可以从聚合 GP 模型的理论一致性可以体现出。并且，GPoE 预测的侧重点为预测的方差上，可以看出其预测均值在一些

数据集上表现并不良好，如 superconductor 数据集和 electric 数据集。考虑 GRBCM 模型，其在 superconductor 数据集、protein 数据集和 wec 数据集上表现优异，但对于一些更大的数据集，其预测结果则相对较差。相反，GPoGRBCM 总是提供了不错的预测结果。这意味着使用双层结构可以帮助模型在百万级的数据集上保持有效的预测一致性，其中数据子集的大小被实际情况限制在10^4以内。在大数据集上，GPoGRBCM 的预测优于 GPoE 的预测，因为考虑 GPoGRBCM 的 GPoE 结构（即第一层）时，它近似了一个更大的、不再被限制在10^4以内的数据子集。

表 4.5 GPoGRBCM 中分区方法、第一层大小、第二层大小的关系。当第二层大小为 250 时，(M1,M2)=(40,160),(80,80),(160,40)；第二层大小为 500 时，(M1,M2)=(40,80),(64,50),(80,40)；第二层大小为 1000 时，(M1,M2)=(20,80),(40,40),(80,20)。

$GPoGRBCM_k$($GPoGRBCM_k$－$GPoGRBCM_k$)			
n/M1/M2	SMSE		
	M1 > M2	M1 ≈ M2	M1 < M2
250	10.3122(+0.9628)	10.1735(+0.8735)	11.2764(+0.1356)
500	9.8378(+0.4402)	9.5327(+0.6083)	9.8006(+0.4084)
1000	9.3939(+0.5625)	9.3604(+0.4193)	**9.2101**(+0.5802)
n/M1/M2	MSLL		
	M1 > M2	M1 ≈ M2	M1 < M2
250	−3.4155(+0.1056)	−3.3786(+0.2442)	−3.2976(+0.3032)
500	−3.4537(+0.0853)	−3.4598(+0.1291)	−3.4147(+0.1625)
1000	**−3.5095**(+0.0632)	−3.4846(+0.0915)	−3.4633(+0.1956)

与其他聚合 GP 模型不同的是，GPoGRBCM 是一个双层的模型，故当比较数据子集大小$n/M1/M2$时，需要关注其第一层数据子集的大小$n/M1$。此时本文同时考虑使用随机分区与聚类分区的 GPoGRBCM 模型，在广泛使用的 electirc 数据集上进行实验。表 4.5 清楚地显示了聚类分区在该实验中具有压倒性的优势。而且，预测的精确性主要取决于数据子集$n/M1/M2$的大小。而第一层数据子集的大小$n/M1$则控制了 GPoGRBCM 模型是更接近于 GPoE 还是 GRBCM，其中当$M1<M2$时模型像 GRBCM 一样修正以提供更优的预测均值，而当$M1>M2$时则如 GPoE 一样注重全局以给出更优的预测方差。

最后，考虑聚合模型对于优化算法的鲁棒性，本文分别比较了最大优化步数为 25、50、100 和 500 的优化算法对模型结构的影响。图 4.7 简明地揭示了对于不同的最大步数，模型 $GPoGRBCM_k$ 有着最优的 SMSE 和 MSLL，且其对最大步数不敏感。

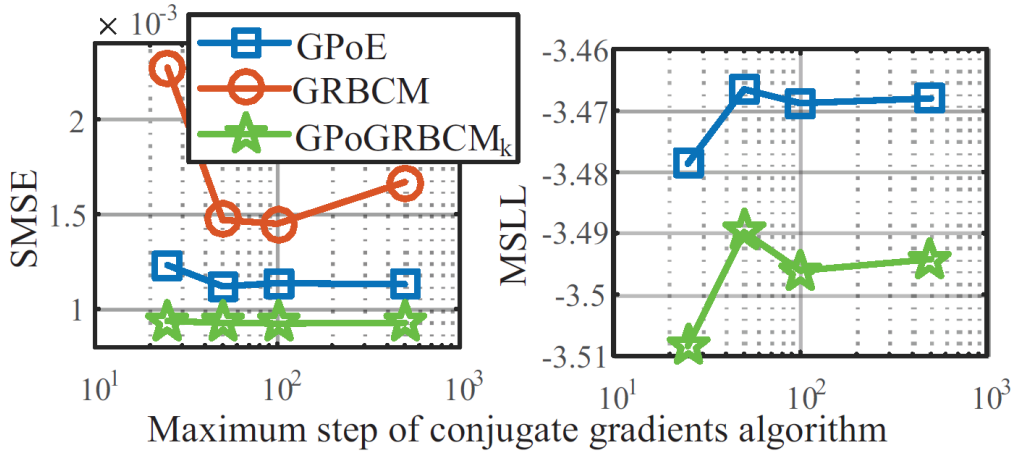

图 4.7　共轭梯度下降算法的最大迭代步数对聚合 GP 模型精度的影响。

4.5 结论与讨论

为了保持模型在大数据集上的有效的一致性，本文从分析典型的聚合 GP 模型的收敛性开始，推导出拥有嵌套双层结构的 GPoGRBCM 模型，其结合了 GPoE 关注全局的能力以及 GRBCM 捕捉局部特征的能力。本文也通过大量的实验来证明 GPoGRBCM 模型的有效性，尤其是当数据集的大小 $n>10^5$ 且聚类分区也被运用到模型中时。此外，当遇到中等数据集时，GPoGRBCM 能够还原到更精确的 GRBCM。

虽然聚合 GP 模型是一类具体的拓展 GP 到大数据集上的方式，但是对于大多的近似模型而言总会涉及到一个问题—“GP 需要完整的数据集吗？”[91]。换言之，我们需要用到多少数据点去描绘或者说是“控制”各种各样复杂的隐函数？如聚合 GP 模型中每个数据子集需要多少个数据点，或稀疏近似 GP 模型中需要多少个归纳点？

5 变分稀疏异方差高斯过程回归中的诱导点选择

在先前的介绍中，已经了解了高斯过程回归模型在各领域都有着广泛的应用，并且知道其优势在于能够度量预测的不确定性。但从模型的设置上看可以发现，噪声方差或者说是观测方差σ_ε^2总被假设为常数，这直观地看并不能满足所有任务与应用的需求，所以需要异方差模型去模拟复杂的数据生成模式，如在金融时间序列领域被广泛应用的经典的 GARCH 模型[158][159]。一个直接的想法是将噪声方差与输入$\boldsymbol{x}$直接挂钩，即之间通过位置去建模噪声方差，但这些做法通常会导致训练与预测难以分析，因为标准的高斯过程回归模型还有一个好处是其分析的结果都有解析形式。

至今为止也有大量的工作致力于解决异方差高斯过程回归（Heteroscedastic GP，HGP）模型中不可解析的边际似然$p(\boldsymbol{y})$和预测分布$p(y_*|\mathcal{D},\boldsymbol{x}_*)$。比起多阶段分别建模隐函数$f$与噪声$\varepsilon$的方法[160][161]，将两者纳入统一框架下的方法会更具吸引力。无论是直接将隐函数与噪声使用两个 GP 来建模[162]，还是间接地非线性组合它们，都需要用到许多近似方法，如 MCMC[165]、最大后验估计[166]、变分推断[70][170][171]、期望传播算法[172][173]、拉普拉斯近似[174]。MCMC 与拉普拉斯近似会更加耗时，不太适合处理大数据；而最大后验估计是点估计，可能会导致模型过拟合；期望传播算法则与变分算法类似，是确定性近似，但该算法并不能保证收敛。相比之下，变分法即保证了收敛性，又适合处理大数据，而且天然地具有抗过拟合的能力。

本部分内容主要关注分布式变分稀疏 HGP（Distributed Variational Sparse HGP，DVSHGP），它同时通过全局近似与局部近似加速运算，其中全局近似指的是使用诱导点方法，局部近似指的是使用聚合模型方法。当诱导点方法被使用时，一个很自然的问题，即是使用多少个诱导点才能够提炼原数据集的绝大多数信息或者说保证高质量的近似呢？从核函数的视角来看，Nystrom 方法已被大量运用于核机器学习模型中[175]，特别是为这些模型提供统计上的保证[176]，如收敛率。有限维近似模型作为基于模型视角的代表，也已被应用到 HGP 中。特别地，Ferrari-Trecate 根据最大的 m 个特征值与σ_ε^2/n的关系来选择最优的维度。随后，变分法也被应用到全局近似当中。Titsias 为挑选诱导点设计了后验的贪婪 EM 算法，David 则通过约束边际似然的上下界（等同于约束 KL 散度）给出了先验的选择策略，即使用$m=\mathcal{O}(\log^D n)$个诱导点去近似 D 维的 SE 核[140][177]。

根据先前的描述，两种建议分别偏向理论与实践。理论方面，主要考虑 KL

散度的上界，因为标准 GP 回归模型关于诱导点的边际似然的上界 L_{up} 与下界 L_{low} 是可以解析的。或者使用对数边际似然直接减去证据下界（Evidence Lower Bound，ELBO）$\log p(\boldsymbol{y})-KL$，得到 KL 散度的解析形式。但对于 HGP，虽然其边际似然及其上界是存在的，如 EUBO[178]和 CUBO[179]，但是都没有解析形式。对于应用而言，添加大量的诱导点直到 ELBO 不再上升是一个比较保守的方法，即使如此，在实践中还得面对什么时候停止添加点和怎么添加点的问题。

对于 DVSHGP，它有着强于标准 GP 的解释能力以及综合全局近似与局部近似的良好拓展性，这驱使本文以该模型为研究对象，想要考虑其收敛率。首先，本文通过比较 VSHGP 和 VHGP，研究了诱导点为该模型带来的影响。基于此，最小化标准核函数矩阵与 Nystrom 近似矩阵差的秩可以被作为一个充分条件来最小化 VSHGP 与 VHGP 模型之间的差别。因此，本文（i）使用 SE 核与 SEARD 核矩阵的迹的上界，给出了诱导点个数选择的先验建议；（ii）拓展了 Titsias 的后验贪婪 EM 算法，使得 DVSHGP 能够迭代地基于迹添加诱导点，以至期望的预测精度。

5.1 分布式变分稀疏异方差高斯过程回归模型回顾

HGP 回归任务可以由 $y(\boldsymbol{x})=f(\boldsymbol{x})+\varepsilon(\boldsymbol{x})$ 表示，其中 $f(\boldsymbol{x})$ 为真实值函数，$\varepsilon(\boldsymbol{x})$ 为观测噪声，服从

$$f(\boldsymbol{x})\sim\mathcal{GP}\left(0,k^{f}(\boldsymbol{x},\boldsymbol{x}')\right),\ \ \varepsilon(\boldsymbol{x})\sim N\left(0,\sigma_{\varepsilon}^{2}(\boldsymbol{x})\right), \tag{5.1}$$

$$\sigma_{\varepsilon}^{2}(\boldsymbol{x})=e^{g(x)},\ \ g(\boldsymbol{x})\sim GP\left(\mu_0,k^{g}(\boldsymbol{x},\boldsymbol{x}')\right) \tag{5.2}$$

其中指数项 $e^{g(x)}$ 保证了方差为正，常数均值 μ_0 提升了对于噪声的建模能力。另外，非稀疏且非变分的 HGP 仅依赖于 GP 的超参数 $\boldsymbol{\theta}=\left\{\boldsymbol{\theta}_f,\mu_0,\boldsymbol{\theta}_g\right\}$，其中 $\boldsymbol{\theta}_f$ 和 $\boldsymbol{\theta}_g$ 由具体的核函数决定，通常可以使用常用的 SEARD 核，同式（2.2）

$$k(\boldsymbol{x},\boldsymbol{x}')=v^{2}\exp\left(-\frac{1}{2}\sum_{i=1}^{D}\frac{(x_i-x_i')^{2}}{l_i^{2}}\right), \tag{5.3}$$

其中 v^2 为输出信号超参数，l_i 为第 i 维的输入尺度超参数。

给定数据集 $\mathcal{D}=\{\boldsymbol{X},\boldsymbol{y}\}$，则有先验

$$p(\boldsymbol{f})=N\left(\boldsymbol{f}\,|\,\boldsymbol{0},\boldsymbol{K}_{nn}^{f}\right),\ \ p(\boldsymbol{g})=N\left(\boldsymbol{g}\,|\,\mu_0\boldsymbol{1},\boldsymbol{K}_{nn}^{g}\right), \tag{5.4}$$

其中 $n\times n$ 矩阵中的元素为 $\left[\boldsymbol{K}_{nn}^{f}\right]_{ij}=k^{f}(\boldsymbol{x}_i,\boldsymbol{x}_j)$ 和 $\left[\boldsymbol{K}_{nn}^{g}\right]_{ij}=k^{g}(\boldsymbol{x}_i,\boldsymbol{x}_j)$。相应地，似然函数有 $p(\boldsymbol{y}\,|\,\boldsymbol{f},\boldsymbol{g})=N(\boldsymbol{y}\,|\,\boldsymbol{f},\boldsymbol{\Sigma}_{\varepsilon})$，其中协方差阵为对角阵，有 $\left[\boldsymbol{\Sigma}_{\varepsilon}\right]_{ii}=e^{g(\boldsymbol{x}_i)}$。

为了加速 HGP，稀疏近似的框架为真实值函数使用了 m 个诱导点，即在输入位置 $\boldsymbol{X}_m$ 上有 $\boldsymbol{f}_m\sim\mathcal{N}\left(\boldsymbol{f}_m\,|\,\boldsymbol{0},\boldsymbol{K}_{mm}^{f}\right)$，相应地对噪声使用类似的 u 个输入输出对 $\{\boldsymbol{X}_u,\boldsymbol{g}_u\}$，有 $\boldsymbol{g}_u\sim\mathcal{N}\left(\boldsymbol{g}_u\,|\,\mu_0\boldsymbol{1},\boldsymbol{K}_{uu}^{g}\right)$。此时，条件分布则可如下表示

$$p(\boldsymbol{f}\mid\boldsymbol{f}_m)=\mathcal{N}\left(\boldsymbol{f}\mid\boldsymbol{K}_{nm}^{f}\left(\boldsymbol{K}_{mm}^{f}\right)^{-1}\boldsymbol{f}_m,\boldsymbol{K}_{nn}^{f}-\boldsymbol{Q}_{nn}^{f}\right),\tag{5.5}$$

$$p(\boldsymbol{g}\mid\boldsymbol{g}_u)=\mathcal{N}\left(\boldsymbol{g}\mid\boldsymbol{K}_{nu}^{g}\left(\boldsymbol{K}_{uu}^{g}\right)^{-1}(\boldsymbol{g}_u-\mu_0\mathbf{1})+\mu_0\mathbf{1},\boldsymbol{K}_{nn}^{g}-\boldsymbol{Q}_{nn}^{g}\right),\tag{5.6}$$

其中 $\boldsymbol{K}_{nm}^{f}=k^{f}(\boldsymbol{X},\boldsymbol{X}_m)$，$\boldsymbol{K}_{nu}^{g}=k^{g}(\boldsymbol{X},\boldsymbol{X}_u)$，$\boldsymbol{Q}_{nn}^{f}=\boldsymbol{K}_{nm}^{f}\left(\boldsymbol{K}_{mm}^{f}\right)^{-1}\boldsymbol{K}_{mn}^{f}$，$\boldsymbol{Q}_{nn}^{g}=\boldsymbol{K}_{nu}^{g}\left(\boldsymbol{K}_{uu}^{g}\right)^{-1}\boldsymbol{K}_{un}^{g}$。

5.1.1 训练

为了解决用于训练的对数边际似然$\log p(\boldsymbol{y})$的不可解析问题，变分法导出了其可解析的下界 ELBO

$$ELBO(q(\boldsymbol{z}))=\int q(\boldsymbol{z})\log\frac{p(\boldsymbol{z},\boldsymbol{y})}{q(\boldsymbol{z})}d\boldsymbol{z}=\log(\boldsymbol{y})-KL(q(\boldsymbol{z})\|p(\boldsymbol{z}\mid\boldsymbol{y})),\tag{5.7}$$

其中$\boldsymbol{z}=\{\boldsymbol{f},\boldsymbol{g},\boldsymbol{f}_m,\boldsymbol{g}_u\}$，$q(\boldsymbol{z})$是变分分布。因为$KL(\cdot\|\cdot)\geq 0$，所以只要$q(\boldsymbol{z})$足够接近精确的后验分布$p(\boldsymbol{z}\mid\boldsymbol{y})$，ELBO 可以作为$\log p(\boldsymbol{y})$的替代品。因此，变分法的目标是使变分分布$q(\boldsymbol{z})$尽可能接近后验$p(\boldsymbol{z}\mid\boldsymbol{y})$，这也等价于以最小化 ELBO 为目标训练超参数。此时，由于真实值函数与噪声的独立性，后验分布及其近似可以分解如下

$$p(\boldsymbol{z}\mid\boldsymbol{y})=p(\boldsymbol{f}\mid\boldsymbol{f}_m)p(\boldsymbol{g}\mid\boldsymbol{g}_u)p(\boldsymbol{f}_m\mid\boldsymbol{y})p(\boldsymbol{g}_u\mid\boldsymbol{y})\tag{5.8}$$

$$p(\boldsymbol{z})=p(\boldsymbol{f}\mid\boldsymbol{f}_m)p(\boldsymbol{g}\mid\boldsymbol{g}_u)q(\boldsymbol{f}_m)q(\boldsymbol{g}_u)\tag{5.9}$$

使用平均场假设，当$q(\boldsymbol{g}_u)$给定时有最优变分分布$q^{*}(\boldsymbol{f}_m)$，反之亦然。一种方法是使用坐标上升法（Coordinate Ascent Variational Inference），迭代地生成局部最优分布。基于梯度的方法则固定了关于$q(\boldsymbol{g}_u)$的最优分布$q^{*}(\boldsymbol{f}_m)$，即优化时只需考虑分布$q(\boldsymbol{g}_u)$。给定变分分布$q(\boldsymbol{g}_u)=\mathcal{N}(\boldsymbol{g}_u\mid\boldsymbol{\mu}_u,\boldsymbol{\Sigma}_u)$，更紧的变分下界可以导出为

$$\begin{aligned}ELBO_{sparse}(q(\boldsymbol{g}_u))=&\log\mathcal{N}(\boldsymbol{y}\mid\mathbf{0},\boldsymbol{Q}_{nn}^{f}+\boldsymbol{R}_g)-0.5\mathrm{Tr}\left[\boldsymbol{R}_g^{-1}\left(\boldsymbol{K}_{nn}^{f}-\boldsymbol{Q}_{nn}^{f}\right)\right]\\&-0.25\mathrm{Tr}\left[\boldsymbol{\Sigma}_g\right]-KL(q(\boldsymbol{g}_u)\|p(\boldsymbol{g}_u))\end{aligned}\tag{5.10}$$

其中$\boldsymbol{R}_g$为$n\times n$的对角阵，元素为$\left[\boldsymbol{R}_g\right]_{ii}=e^{\left[\boldsymbol{\mu}_g\right]_i-\left[\boldsymbol{\Sigma}_g\right]_{ii}/2}$，根据$p(\boldsymbol{g}\mid\boldsymbol{y})$的近似$q(\boldsymbol{g})=\int p(\boldsymbol{g}\mid\boldsymbol{g}_u)q(\boldsymbol{g}_u)d\boldsymbol{g}_u$有均值和方差

$$\boldsymbol{\mu}_g=\boldsymbol{K}_{nu}^{g}\left(\boldsymbol{K}_{uu}^{g}\right)^{-1}(\boldsymbol{\mu}_u-\mu_0\mathbf{1})+\mu_0\mathbf{1},\tag{5.11}$$

$$\boldsymbol{\Sigma}_g=\boldsymbol{K}_{nn}^{g}-\boldsymbol{Q}_{nn}^{g}+\boldsymbol{K}_{nu}^{g}\left(\boldsymbol{K}_{uu}^{g}\right)^{-1}\boldsymbol{\Sigma}_u\left(\boldsymbol{K}_{uu}^{g}\right)^{-1}\boldsymbol{K}_{un}^{g}.\tag{5.12}$$

进一步，一个$n\times n$的对角矩阵$\boldsymbol{\Lambda}_{nn}$可被用来降低变分参数的数量，即从原来均值和协方差需要$u+u(u+1)/2$个参数降为 n 个，该方法在需要大量诱导点的任务中效果明显。此时由驻点方程$\partial ELBO_{sparse}/\partial\boldsymbol{\mu}_u=0$和$\partial ELBO_{sparse}/\partial\boldsymbol{\Sigma}_u=0$可得最优参数

$$\boldsymbol{\mu}_u = \boldsymbol{K}_{un}^g\left(\boldsymbol{\Lambda}_{nn} - 0.5\boldsymbol{I}\right)\mathbf{1} + \mu_0\mathbf{1}, \tag{5.13}$$

$$\boldsymbol{\Sigma}_u^{-1} = \left(\boldsymbol{K}_{uu}^g\right)^{-1} + \left(\boldsymbol{K}_{uu}^g\right)^{-1}\boldsymbol{K}_{un}^g\boldsymbol{\Lambda}_{nn}\boldsymbol{K}_{nu}^g\left(\boldsymbol{K}_{uu}^g\right)^{-1}. \tag{5.14}$$

至此，只要通过优化 ELBO 来联合训练所有的超参数 $\boldsymbol{\Theta} = \left\{\boldsymbol{\Lambda}_{nn}, \boldsymbol{\theta}, \boldsymbol{X}_m, \boldsymbol{X}_u\right\}$，之后就可以直接进行预测。

特别地，当我们令 $\boldsymbol{f} = \boldsymbol{f}_m$ 和 $\boldsymbol{g} = \boldsymbol{g}_u$ 时，非稀疏版本的 ELBO 可以表示为

$$ELBO_{full}\left(q(\boldsymbol{g})\right) = \log\mathcal{N}\left(\boldsymbol{y}\,|\,\mathbf{0}, \boldsymbol{K}_{nn}^f + \boldsymbol{R}\right) - 0.25\mathrm{Tr}\left[\boldsymbol{\Sigma}\right] - KL\left(q(\boldsymbol{g})\,\|\,p(\boldsymbol{g})\right) \tag{5.15}$$

其中对角阵 $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ 有元素 $\left[\boldsymbol{R}\right]_{ii} = e^{[\boldsymbol{\mu}]_i - [\boldsymbol{\Sigma}]_{ii}/2}$，并且相应有

$$\boldsymbol{\mu} = \boldsymbol{K}_{nn}^g\left(\boldsymbol{\Lambda}_{nn} - 0.5\boldsymbol{I}\right)\mathbf{1} + \mu_0\mathbf{1}, \tag{5.16}$$

$$\boldsymbol{\Sigma}^{-1} = \left(\boldsymbol{K}_{nn}^g\right)^{-1} + \boldsymbol{\Lambda}_{nn}. \tag{5.17}$$

5.1.2 预测

虽然预测分布 $p\left(y_* \,|\, \mathcal{D}, \boldsymbol{x}_*\right)$ 不能直接由封闭式的推断得出解析形式，但其均值 μ_* 与方差 σ_*^2 可以使用近似方法得到相应的解析形式。令 $f_* = f\left(\boldsymbol{x}_*\right)$，真实值函数的近似后验分布有 $q\left(f_*\right) = \int p\left(f_* \,|\, \boldsymbol{f}_m\right)q^*\left(\boldsymbol{f}_m\right)d\boldsymbol{f}_m = N\left(f_* \,|\, \mu_{f_*}, \sigma_{f_*}^2\right)$，其中

$$\mu_{f_*} = \boldsymbol{k}_{*m}^f\boldsymbol{K}_R^{-1}\boldsymbol{K}_{mn}^f\boldsymbol{R}_g^{-1}\boldsymbol{y}, \tag{5.18}$$

$$\sigma_{f_*}^2 = k_{**}^f - \boldsymbol{k}_{*m}^f\left(\boldsymbol{K}_{mm}^f\right)^{-1}\boldsymbol{k}_{m*}^f + \boldsymbol{k}_{*m}^f\boldsymbol{K}_R^{-1}\boldsymbol{k}_{m*}^f, \tag{5.19}$$

其中 $\boldsymbol{K}_R = \boldsymbol{K}_{mn}^f\boldsymbol{R}_g^{-1}\boldsymbol{K}_{mn}^f + \boldsymbol{K}_{mm}^f$。令 $g_* = g\left(\boldsymbol{x}_*\right)$，则对于噪声同样会有相似的预测分布 $q\left(g_*\right) = \int p\left(g_* \,|\, \boldsymbol{g}_u\right)q\left(\boldsymbol{g}_u\right)d\boldsymbol{g}_u = N\left(g_* \,|\, \mu_{g_*}, \sigma_{g_*}^2\right)$，其中

$$\mu_{g_*} = \boldsymbol{k}_{*u}^g\left(\boldsymbol{K}_{uu}^g\right)^{-1}\left(\boldsymbol{\mu}_u - \mu_0\mathbf{1}\right) + \mu_0\mathbf{1}, \tag{5.20}$$

$$\sigma_{g_*}^2 = k_{**}^g - \boldsymbol{k}_{*u}^g\left(\boldsymbol{K}_{uu}^g\right)^{-1}\boldsymbol{k}_{u*}^g + \boldsymbol{k}_{*u}^g\left(\boldsymbol{K}_{un}^g\boldsymbol{\Lambda}_{nn}^{-1}\boldsymbol{K}_{nu}^g + \boldsymbol{K}_{uu}^g\right)^{-1}\boldsymbol{k}_{u*}^g. \tag{5.21}$$

给定似然函数 $p\left(y_* \,|\, f_*, g_*\right) = \mathcal{N}\left(y_* \,|\, f_*, e^{g_*}\right)$，其近似的后验分布可以表示为 $q\left(y_*\right) = \int p\left(y_* \,|\, f_*, g_*\right)q\left(f_*\right)q\left(g_*\right)df_*dg_*$，并且该分布的均值与方差可以使用高斯赫米特正交法写成解析形式，有

$$\mu_* = \mu_{f_*},\ \ \sigma_*^2 = \sigma_{f_*}^2 + e^{\mu_{g_*} + \sigma_{g_*}^2/2}。 \tag{5.22}$$

5.2 最优诱导点

理想情况下，被选择的诱导点的个数需要满足使 KL 散度非常小的能力。然而，不可解析的 $\log p(\boldsymbol{y})$ 及其上界 L_{up}（如 $EUBO = \int p(\boldsymbol{z}|\boldsymbol{y})\log\frac{p(\boldsymbol{z},\boldsymbol{y})}{q(\boldsymbol{z})}d\boldsymbol{z}$ 或 $CUBO_\alpha = \frac{1}{\alpha}\int q(\boldsymbol{z})\left[\log\frac{p(\boldsymbol{z},\boldsymbol{y})}{q(\boldsymbol{z})}\right]^\alpha d\boldsymbol{z}$）使得无法直接计算 $KL = \log p(\boldsymbol{y}) - ELBO$ 或其上界

$KL \le L_{up} - ELBO$。为了评估诱导点的影响，本文则通过分解 KL 散度来间接考虑。

一般的，可以考虑在任意集合 $\boldsymbol{Z}$ 上的后验

$$p(\boldsymbol{Z}|\boldsymbol{y}) = \int p(\boldsymbol{Z}|\boldsymbol{f},\boldsymbol{f}_m,\boldsymbol{g},\boldsymbol{g}_u)\,p(\boldsymbol{f}|\boldsymbol{f}_m,\boldsymbol{y})\,p(\boldsymbol{f}_m|\boldsymbol{y})\,p(\boldsymbol{g}|\boldsymbol{g}_u,\boldsymbol{y})\,p(\boldsymbol{g}_u|\boldsymbol{y})\,d\boldsymbol{f}d\boldsymbol{f}_m d\boldsymbol{g}d\boldsymbol{g}_u \tag{5.23}$$

其中 $\boldsymbol{Z}$ 可以被看作 $\boldsymbol{z}$ 的广义形式，或是在没有已知点位置的预测形式。假设 $\boldsymbol{f}_m$ 和 $\boldsymbol{g}_u$ 分别是 $\boldsymbol{f}$ 和 $\boldsymbol{g}$ 的充分统计量，即有 $p(\boldsymbol{Z}|\boldsymbol{f},\boldsymbol{f}_m,\boldsymbol{g},\boldsymbol{g}_u) = p(\boldsymbol{Z}|\boldsymbol{f}_m,\boldsymbol{g}_u)$。则近似后验可以写为

$$q(\boldsymbol{Z}) = \int p(\boldsymbol{Z}|\boldsymbol{f}_m,\boldsymbol{g}_u)\,p(\boldsymbol{f}|\boldsymbol{f}_m)\,q(\boldsymbol{f}_m)\,p(\boldsymbol{g}|\boldsymbol{g}_u)\,q(\boldsymbol{g}_u)\,d\boldsymbol{f}d\boldsymbol{f}_m d\boldsymbol{g}d\boldsymbol{g}_u \tag{5.24}$$

其中 $p(\boldsymbol{f}|\boldsymbol{f}_m,\boldsymbol{y}) = p(\boldsymbol{f}|\boldsymbol{f}_m)$ 可被理解为图模型中使用 $\boldsymbol{y}$ 推断 $\boldsymbol{f}_m$ 推断 $\boldsymbol{f}$ 的直线过程，此时给定中间的 $\boldsymbol{f}_m$ 有 $\boldsymbol{y}$ 与 $\boldsymbol{f}$ 条件独立，同样的有 $p(\boldsymbol{g}|\boldsymbol{g}_u,\boldsymbol{y}) = p(\boldsymbol{g}|\boldsymbol{g}_u)$。根据后验分布及其近似分布的结构差异，可以将整体的训练分为两部分，即诱导点的数量是否充分以至于满足充分统计量的假设，和变分参数是否满足 $q(\boldsymbol{f}_m) = p(\boldsymbol{f}_m|\boldsymbol{y})$ 和 $q(\boldsymbol{g}_u) = p(\boldsymbol{g}_u|\boldsymbol{y})$。

回顾从 $ELBO_{full}$ 到 $ELBO_{sparse}$ 的过程，需要训练的部分也多出了关于诱导点的参数。因此，本文希望使用 $\log p(\boldsymbol{y})$、$ELBO_{full}$ 和 $ELBO_{sparse}$ 来评估相应参数的作用。首先，$ELBO_{sparse} \le ELBO_{full}$ 可由 $ELBO_{sparse}$ 关于诱导点数量的非严格单调性得出。

命题 5.1：令 $\left(\boldsymbol{X}_{\langle m\rangle},\boldsymbol{f}_{\langle m\rangle}\right)$ 和 $\left(\boldsymbol{X}_{\langle u\rangle},\boldsymbol{g}_{\langle u\rangle}\right)$ 为相应的诱导点集合，其中 $\langle\cdot\rangle$ 表示相应数量的诱导点的指示集合，添加任意的诱导点 $\langle\Delta m\rangle \subseteq \langle n\rangle - \langle m\rangle$ 或 $\langle\Delta u\rangle \subseteq \langle n\rangle - \langle u\rangle$ 将不会减少 $ELBO_{sparse}$，其中 $\langle n\rangle = \{1,\ldots,n\}$。

证明：在添加数据点前，最优的后验近似 $q(\boldsymbol{z})$ 可被写为 $p\left(\boldsymbol{f}|\boldsymbol{f}_{\langle m\rangle}\right)p\left(\boldsymbol{g}|\boldsymbol{g}_{\langle u\rangle}\right)q^*\left(\boldsymbol{f}_{\langle m\rangle}\right)q^*\left(\boldsymbol{g}_{\langle u\rangle}\right) = p\left(\boldsymbol{f}_{\langle n\rangle-\langle m\rangle}|\boldsymbol{f}_{\langle m\rangle}\right)p\left(\boldsymbol{g}_{\langle n\rangle-\langle u\rangle}|\boldsymbol{g}_{\langle u\rangle}\right)q^*\left(\boldsymbol{f}_{\langle m\rangle}\right)q^*\left(\boldsymbol{g}_{\langle u\rangle}\right)$，其中 $p\left(\boldsymbol{f}|\boldsymbol{f}_{\langle m\rangle}\right) = p\left(\boldsymbol{f}_{\langle n\rangle-\langle m\rangle}|\boldsymbol{f}_{\langle m\rangle}\right)$ 因为 $\boldsymbol{f}$ 是 $\boldsymbol{f}_m$ 的完整版本，对 g 同理。对于 f，添加 Δm 个点后有 $p\left(\boldsymbol{f}_{\langle n\rangle-\langle m\rangle}|\boldsymbol{f}_{\langle m\rangle}\right)q^*\left(\boldsymbol{f}_{\langle m\rangle}\right) = p\left(\boldsymbol{f}_{\langle n\rangle-\langle m\rangle\cup\langle\Delta m\rangle}|\boldsymbol{f}_{\langle\Delta m\rangle},\boldsymbol{f}_{\langle m\rangle}\right)p\left(\boldsymbol{f}_{\langle\Delta m\rangle}|\boldsymbol{f}_{\langle m\rangle}\right)$ $\times q^*\left(\boldsymbol{f}_{\langle m\rangle}\right)$，其中局部最优项 $p\left(\boldsymbol{f}_{\langle\Delta m\rangle}|\boldsymbol{f}_{\langle m\rangle}\right)q^*\left(\boldsymbol{f}_{\langle m\rangle}\right)$ 可被全局最优项 $q^*\left(\boldsymbol{f}_{\langle\Delta m\rangle},\boldsymbol{f}_{\langle m\rangle}\right)$ 替代。同理可得对于 g 的结论，所以添加点诱导点会使 $ELBO_{sparse}$ 非严格单调递增。□

至此，我们有 $ELBO_{sparse} \le ELBO_{full} \le \log p(\boldsymbol{y})$。考虑 Θ，可以清楚看出同样有三种不同功能的超参数。具体地，$ELBO_{sparse}$ 和 $ELBO_{full}$ 之间的差异由诱导点主导，$ELBO_{full}$ 和 $\log p(\boldsymbol{y})$ 之间的差异主要由变分参数主导，而 $\log p(\boldsymbol{y})$ 的高度则主要由 GP 超参数 $\boldsymbol{\theta}$ 决定。给定变分参数与 GP 超参数，为了学习最优的诱导

点最小化 $ELBO_{sparse}$ 与 $\log p(\boldsymbol{y})$ 间的 KL 散度，等价于最小化 $ELBO_{sparse}$ 和 $ELBO_{full}$ 之间的差异。因此，$ELBO_{full}-ELBO_{sparse}$ 可被当作 $KL=\log p(\boldsymbol{y})-ELBO_{sparse}$ 的固定变分参数的版本，这使得我们给出了一个充分条件使得 $ELBO_{full}-ELBO_{sparse}$ 很小。

命题 5.2：令 $T_f=\mathrm{Tr}\left[\boldsymbol{K}_{nn}^f-\boldsymbol{Q}_{nn}^f\right]$，$T_g=\mathrm{Tr}\left[\boldsymbol{K}_{nn}^g-\boldsymbol{Q}_{nn}^g\right]$，如果 $T_f=T_g=0$，则 $ELBO_{full}-ELBO_{sparse}=0$。

证明：固定关于 g 的变量，则可得到 $ELBO_{sparse}$ 关于 f 的同方差的版本，当 $\mathrm{Tr}\left[\boldsymbol{K}_{nn}^f-\boldsymbol{Q}_{nn}^f\right]=0$ 且 $\boldsymbol{K}_{nn}^f=\boldsymbol{Q}_{nn}^f$ 时有最大值 $\log\mathcal{N}\left(\boldsymbol{y}\,|\,\boldsymbol{0},\boldsymbol{K}_{nn}^f+\boldsymbol{R}_g\right)$。记 $\boldsymbol{K}_{nn}^f\succeq\boldsymbol{Q}_{nn}^f$ 表示矩阵 $\boldsymbol{K}_{nn}^f-\boldsymbol{Q}_{nn}^f$ 为对称半正定矩阵，因为 $\boldsymbol{Q}_{nn}^f$ 是 $\boldsymbol{K}_{nn}^f$ 的 Nystrom 表示，故有 $\boldsymbol{K}_{nn}^f\succeq\boldsymbol{Q}_{nn}^f$，此时 $\mathrm{Tr}\left[\boldsymbol{K}_{nn}^f-\boldsymbol{Q}_{nn}^f\right]=0$ 可直接得出 $\boldsymbol{K}_{nn}^f=\boldsymbol{Q}_{nn}^f=\boldsymbol{K}_{mm}^f$。此时固定 f，并且给定 $\mathrm{Tr}\left[\boldsymbol{K}_{nn}^g-\boldsymbol{Q}_{nn}^g\right]=0$，则有 $\boldsymbol{\mu}_u=\boldsymbol{\mu}_g=\boldsymbol{\mu}$，$\boldsymbol{\Sigma}_u=\boldsymbol{\Sigma}_g=\boldsymbol{\Sigma}$，$\boldsymbol{K}_{nn}^g=\boldsymbol{K}_{uu}^g$，可以导出结果 $KL\left(q(\boldsymbol{g}_u)\|\,p(\boldsymbol{g}_u)\right)=KL\left(q(\boldsymbol{g})\|\,p(\boldsymbol{g})\right)$ 和 $\boldsymbol{R}_g=\boldsymbol{R}$。考虑全局最优的条件，因为 f 与 g 是独立的，我们则可以同时满足 $\mathrm{Tr}\left[\boldsymbol{K}_{nn}^f-\boldsymbol{Q}_{nn}^f\right]=0$ 和 $\mathrm{Tr}\left[\boldsymbol{K}_{nn}^g-\boldsymbol{Q}_{nn}^g\right]=0$。此时我们有结果 $ELBO_{full}-ELBO_{sparse}=0$，这也意味着诱导点的数量是足够的。□

5.3 诱导点选择策略

5.3.1 先验策略

通常，$\boldsymbol{K}_{mm}^f$ 包含了 $\boldsymbol{K}_{nn}^f$ 中最大的 m 个特征值 λ，并且 $\boldsymbol{K}_{mm}^f$ 可被直接当作相应的特征向量，对 g 同理。依此，$\boldsymbol{Q}_{nn}^f$ 与 $\boldsymbol{Q}_{nn}^g$ 则分别是 $\boldsymbol{K}_{nn}^f$ 与 $\boldsymbol{K}_{nn}^g$ 的最优 m 阶与 u 阶近似，此时有 $T_f=\sum_{m+1}^{n}\lambda_i\left(\boldsymbol{K}_{nn}^f\right)$ 和 $T_g=\sum_{u+1}^{n}\lambda_i\left(\boldsymbol{K}_{nn}^g\right)$。直接分析核矩阵的谱性质是有吸引力的，因为只需计算相应最大的 m 与 u 个特征值即可，花费 $\mathcal{O}\left(n^2m+n^2u\right)$，如使用 PCA。而且在实际应用中，我们通常在训练之前是不知道最优的核超参数的，并且使用少量的诱导点时会得出局部最优的核超参数可能它与最优的核超参数相距甚远。幸运的是，可以使用核算子 $\mathcal{K}^H$ 来表示 $\boldsymbol{K}_{nn}^H$ 的极限性质，有 $\frac{1}{n}\boldsymbol{K}_{nn}^H\overset{n\to\infty}{\to}\mathcal{K}^H$。令 $\left(\mathbb{R}^D,\mathcal{F},P_H\right)$ 为概率空间，则核算子 $\mathcal{K}^H:L^2\left(\mathbb{R}^D,\mathcal{F},P_H\right)\to L^2\left(\mathbb{R}^D,\mathcal{F},P_H\right)$ 定义为

$$\mathcal{K}^H g(\boldsymbol{x})=\int g(\boldsymbol{x})k^H(\boldsymbol{x},\boldsymbol{x}')dP_H(\boldsymbol{x}') \tag{5.25}$$

其中核函数是连续且有界的。此时核算子的特征值与特征函数对可按特征值的大小 $\lambda_1\geq\lambda_2\geq\cdots\geq0$ 排列为 $\left\{\lambda_j,\phi_j\right\}_{j=1}^{\infty}$。然后考虑核函数的 Mercer-Hilbert 展开，有

$$k^H(\boldsymbol{x},\boldsymbol{x}')=\sum_{i=1}^{\infty}\lambda_j\phi_j(\boldsymbol{x})\phi_j(\boldsymbol{x}') \tag{5.26}$$

这使得

$$\left[\boldsymbol{K}_{nn}^{H}-\boldsymbol{Q}_{nn}^{H}\right]_{ii}=\sum_{j=h+1}^{\infty}\lambda_j\phi_j^2\left(\boldsymbol{x}_i\right) \tag{5.27}$$

对迹求期望，使用上式求和的形式，有

$$\mathbb{E}_X\left[T_H\right]=n\sum_{j=h+1}^{\infty}\lambda_j\mathbb{E}_{\boldsymbol{x}_i}\left[\phi_j^2\left(\boldsymbol{x}_i\right)\right]=n\sum_{j=h+1}^{\infty}\lambda_j \tag{5.28}$$

使用马尔科夫不等式，其中概率令为$1-\delta_H$，则迹的上界可以表示为

$$T_H\le n\sum_{j=h+1}^{\infty}\lambda_j\Big/\delta_H \tag{5.29}$$

当核函数为一维 SE 核的时候，令其超参数为$\boldsymbol{\theta}_H=\left(v_H^2,l_H^2\right)$，以及输入密度为高斯的 $p_H\left(x\right)=P_H'\left(x\right)\sim N\left(0,\sigma_H^2\right)$，则 $\mathcal{K}^H$ 的第 j 个特征值可被计算为 $\lambda_j=v_H^2\sqrt{2a/A}B^{h-1}$，其中 $a=1\Big/\left(4v_H^2\right)$，$b=1\Big/\left(2l_H^2\right)$，$c=\sqrt{a^2+2ab}$，$A=a+b+c$，$B=b/A$。为了计算迹的上界，则级数可以先计算为

$$\sum_{j=h+1}^{\infty}\lambda_j=\frac{v_H^2\sqrt{2a}}{\left(1-B\right)\sqrt{A}}B^h \tag{5.30}$$

当$n\to\infty$时，选择$h=\mathcal{O}\left(\log n\right)$作为诱导点数量可保证$T_H=0$。

对于 D 维情况，不失一般性可以考虑 SEARD 核的各项同性情况，因为特征值可以使用各向异性情况最短输入尺度与最大输入分布的方差来约束。此时有特征值$\lambda_j=\left(2a/A\right)^{D/2}B^{j'}$，其中$j'$为与$j$不同的整数，该等式成立因为 D 维的核函数与输入分布为其一维的 D 次乘积。使用 Burt 等（2020）[177]中的命题 21，可以得到

$$\sum_{j=h+1}^{\infty}\lambda_j=\mathcal{O}\left(h\exp\left(-\alpha h^{1/D}\right)\right) \tag{5.31}$$

其中$\alpha=-\log B>0$，且$h\ge D^D/\alpha+D-1$。类似的，当$n\to\infty$可选择$h=\mathcal{O}\left(\log^D n\right)$使得迹$T_H$足够小，即使用$\mathcal{O}\left(\log^D n\right)$个诱导点对于 VSHGP 是足够的。

5.3.2 后验策略

然而在实际应用中，先验策略只能提供一个大致的选取思路，但很难确定一个真正有效的值以供使用。在本小节中，本文提出了一个后验的贪婪 EM 算法，以便于在观测到数据之后选择诱导点的数量及其位置。

算法 5.1 描述了如何使用命题 5.2 来迭代地选择诱导点。该算法基于(D)VSHGP 模型的多阶段推断，在 E 步时添加诱导点，在 M 步时更新超参数。虽然变分法能够保证连续的收敛性，但该多阶段推断问题因为超参数与诱导点的变化所以是非凸的。因此，重新初始化步骤被添加与 E 步与 M 步之间，防止超参数落入局部最优。最后考虑用户自定义的终止条件Δ。令Ψ为$ELBO_{sparse}^{\tau}$的数量级，一个建议是将终止条件与该数量级挂钩，如使用$\Delta=\left(\Delta m+\Delta u\right)\left(\Psi-3\right)$。至

此，用于 VSHGP 的算法 5.1 已介绍完毕，接下来则介绍 3 个将算法 5.1 应用于 DVSHGP 的注意事项。（a）相同的 Δm 和 Δu 对于每个子模型而言是不合理的，因为数据子集越大时需要的诱导点越多，如对第 k 个子模型令 $\Delta m_k = 0.2\Delta n_k$。（b）对于每个子模型而言，计算迹时的花费可由 $\mathcal{O}\left(n^2\left(m+\Delta m\right)^2 + n^2\left(u+\Delta u\right)^2\right)$ 削减到 $\mathcal{O}\left(n_k^2\left(m+\Delta m\right)^2 + n_k^2\left(u+\Delta u\right)^2\right)$。（c）什么时候可以为每个子模型设置早停标准以加速运算呢？实际上我们很难为一个子模型选择一个合适的 Δ_k，因为数据量与数据模型都会影响该结果。但我们可以尝试着使用迹 T_f^i 和 T_g^i 设置早停，如每个子模型满足 $\min\left\{T_f^i\right\}_{i=1}^{n_k} \le 1$ 和 $\min\left\{T_g^i\right\}_{i=1}^{n_k} \le 1$ 时停止添加点。

算法 5.1　贪婪选择诱导点的 EM 算法伪代码

Algorithm 1 EM algorithm of greedy selecting inducing points

Input: The training dataset $\{\boldsymbol{X}, \boldsymbol{y}\}$, initial hyperparameters $\boldsymbol{\Theta} = \left\{\boldsymbol{\Lambda}_{nn}, \boldsymbol{\theta}, \boldsymbol{X}_m, \boldsymbol{X}_u\right\}$ with the number of inducing points $\{m, u\}$, the number of points expected to increase in each iteration $\{\Delta m, \Delta u\}$, and stop criterion Δ.

Output: Optimal hyperparameters $\boldsymbol{\theta}^\star$ with the optimal number of inducing points $\left\{m^\star, u^\star\right\}$.

repeat
  Initialization.
  for $\tau = 1$ **to** t **do**
    if $\tau = 1$ **then**
      Train the locally optimal hyperparameters $\boldsymbol{\Theta}_\tau = \left\{\boldsymbol{\Lambda}_{nn}^\tau, \boldsymbol{\theta}, \boldsymbol{X}_m^\tau, \boldsymbol{X}_u^\tau\right\}$ via maximizing the $\text{ELBO}_{sparse}^\tau$ based on CGD with the maximum number of evaluations ς.
    else
      E step: Compute the traces $\boldsymbol{T}_f^i$ and $\boldsymbol{T}_g^i$ for n augmented inducing sets $\left\{\boldsymbol{X}_m^\tau, \boldsymbol{x}_i\right\}$ and $\left\{\boldsymbol{X}_u^\tau, \boldsymbol{x}_i\right\}$ respectively. Then select first Δm and Δu points that minimize the traces, we have new inducing sets $\boldsymbol{X}_m^{\tau+0.5} = \left\{\boldsymbol{X}_m, \boldsymbol{X}_{\Delta m}\right\}$ and $\boldsymbol{X}_u^{\tau+0.5} = \left\{\boldsymbol{X}_u, \boldsymbol{X}_{\Delta u}\right\}$.

      M step: Train the locally optimal hyperparameters with re-initialization $\boldsymbol{\Theta}_{\tau+1} = \left\{\boldsymbol{\Lambda}_{nn}, \boldsymbol{\theta}, \boldsymbol{X}_m^{\tau+0.5}, \boldsymbol{X}_u^{\tau+0.5}\right\}$ via maximizing the $\text{ELBO}_{sparse}^{\tau+1}$ based on CGD with the maximum number of evaluations ς.
    end if
  end for
until $\text{ELBO}_{sparse}^{\tau+1} - \text{ELBO}_{sparse}^{\tau} \le \Delta$

5.4 数值实验

本节内容主要关注算法 5.1 应用于 VSHGP 与 DVSHGP 的有效性与高效性。

这些实验主要基于（i）(D)VSHGP 的预测能力与诱导点数量的关系，（ii）算法 5.1 的细节导致其对结果的影响。

本文使用 GPML 工具箱及出版文章中的代码[27]实现(D)VSHGP、(D)VSHGP$_{EM}$、(D)VHGP 及一些分布式 GP 模型，环境为 8GB RAM 和 3.4GHz 四核的个人电脑。所有核函数使用的是 SEARD 核，并且遵循 Liu 等（2021）[70]的超参数初始化过程。对于现实数据集，输入与输出都加以标准化至均值为零、方差为一。最后，SMSE 与 MSLL 被用于评估模型的预测结果，其值越小说明模型结果越好。

5.4.1 玩具数据集

为了可视化诱导点的作用，本文使用 Liu 等（2021）[70]已使用过的 1 维函数

$$y(x)=\mathrm{sinc}(x)+\varepsilon,\quad x\in[-10,10] \tag{5.32}$$

其中异方差噪声有 $\varepsilon\sim\mathcal{N}\left(0,\sigma_\varepsilon^2(x)\right)$，$\sigma_\varepsilon(x)=0.05+0.2\left(1+\sin(2x)\right)/\left(1+e^{-0.2x}\right)$。训练数据与测试数据随机从上式中抽样得。首先，VSHGP 的 CGD 算法使用了最大 200 步迭代，即 $\varsigma=200$，并且两个模型分别使用了 m=u=12 和 m=u=25。图 5.1 说明太少的数据点并不能总结整个数据集的所有信息。虽说 12 个诱导点已经足够模拟变化不大的真实值函数 f，但是比较明显能看出 12 个诱导点并无法满足模拟快速变化的噪声的需要。极端情况下，当诱导点数量过少时，HGP 会退化为 GP。

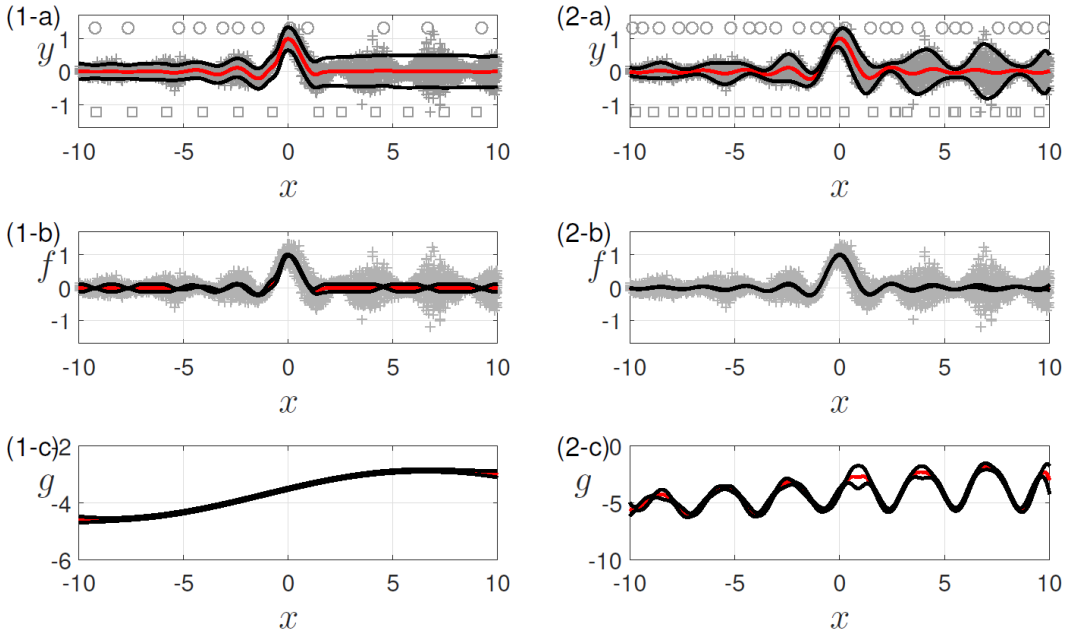

图 5.1 两个使用不同数量诱导点的 VSHGP 模型在玩具数据集上的表现。子图 1 和子图 2 分别表示使用诱导点数量为 m=u=12 和 m=u=25 的模型结果，其中关于 f 的诱导点位置为上方的圆圈，关于 g 的诱导点位置为下方的方块。子图 a-c 分别表示 y、f、g 的预测结果。

[27] https://github.com/LiuHaiTao01

为了显示算法 5.1 作用于具有很少的初始诱导点的情况是有效的，对于该玩具模型，本文令 $m=u=\Delta m=\Delta u=1$，并设置 EM 算法的迭代次数为 25 次，中间使用的 CGD 算法有最大迭代次数 $\varsigma=100$。图 2 则显示了根据迹添加诱导点并且使用重新初始化步骤是可行的。从图 5.2(1-a)中可以看出，当我们根据最大化 T_f 与 T_g 选择诱导点是无效的，可以看出关于噪声的诱导点聚集在一起，并未起到总结全体数据集的作用。从图 5.2(1-b)可以看出，根据该策略添加诱导点的方式，难以实现从同方差到异方差的跨越。图 5.2(2-a)则看出，虽然诱导点的分布是均匀的，但还是难以实现同方差到异方差的跨越，因为在初始的 EM 算法迭代过程中，GP 超参数已经适应了同方差的设定，陷入了局部最优解当中，即我们所知对数边际似然是非凸的。这就需要如重新初始化 GP 超参数的步骤来规避该问题，但方法不是唯一的，且重新初始化也会造成一些问题，即 EM 算法中每个迭代需要花大量的时间训练这些超参数，并且当诱导点过多而 CGD 迭代次数过少时很容易导致添加点后的模型没有添加点前的好。

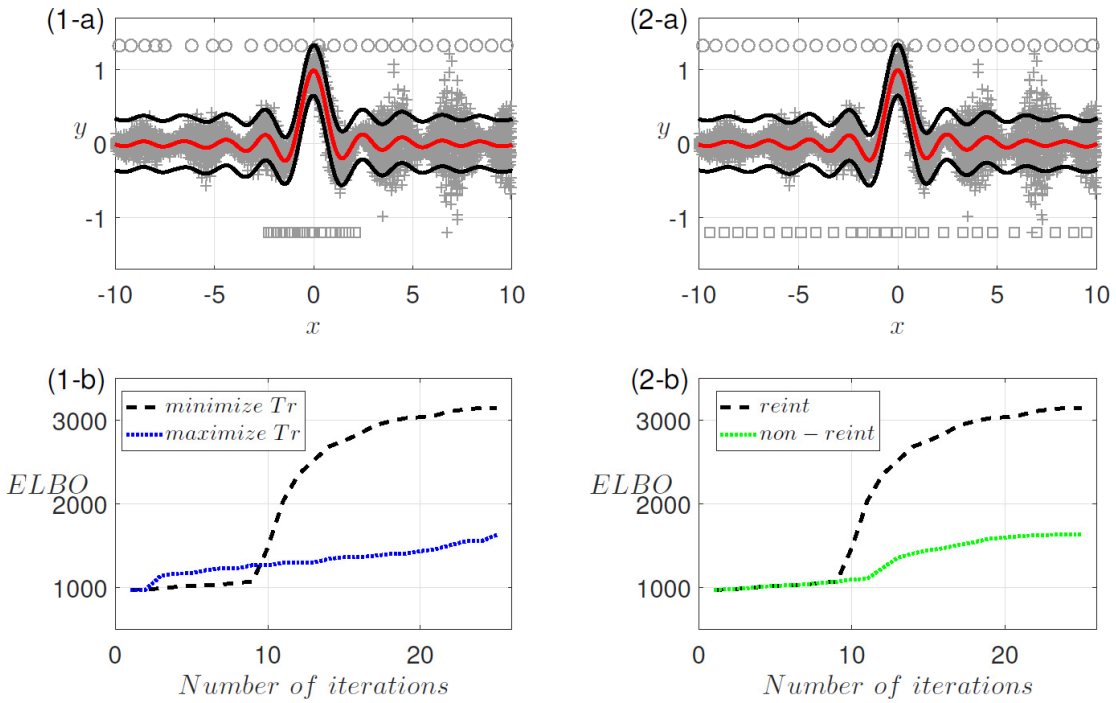

图 5.2　算法 5.1 的细节对玩具数据集上应用的影响。子图 1 显示的是选择诱导点步骤的影响，子图 2 显示的是重新初始化 GP 参数的影响。诱导点的位置在子图 a 当中显示，每次迭代的时候 ELBO 的改变在子图 b 中显示。注：作为对比的 VSHGP 模型可参考图 5.1 的子图 a。

最后，为了探索具体的停止标准Δ对结果的影响，本文将 EM 算法迭代的上限提升到了 50 次，并且选用四个不同的停止标准 $\psi-1$、$\psi-2$、$\psi-3$、$\psi-4$。毫无疑问的是，停止标准设置的过大时，添加诱导点的过程会过早停止，以至于 EM 算法并不能达到很好的效果。而对于简单的问题来说，将停止标准设置的很小时，也不会出现需要过多迭代但对模型提升不大的情况。实验结果可由图 5.3

看到，至少一个小的停止标准可以使 EM 算法将 VSHGP_{EM} 还原为 VSHGP，且该实验并未显示过小的停止标准会使算法 5.3 产生添加多余诱导点的操作。

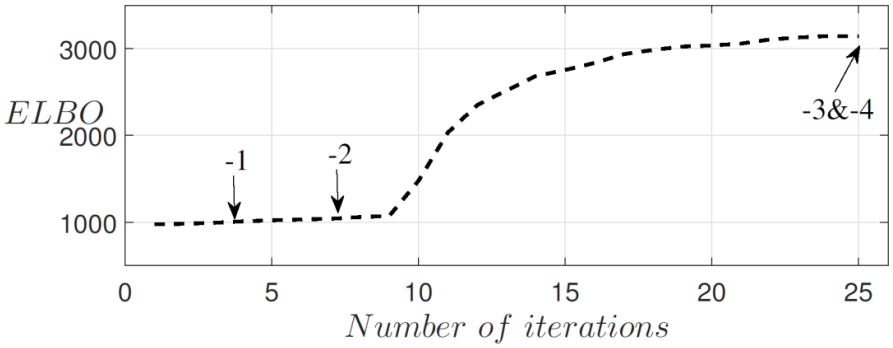

图 5.3 根据不同的停止准则，EM 算法在玩具数据集上停止时的迭代步数

5.4.2 蛋白质数据集

本节主要说明一些算法 5.1 应用于真实数据集时的细节，以 9 维的蛋白质的理化性质数据集[152]为例。首先，随机将数据集分为 36584 个训练样本及 9146 个测试样本。对于本章中 DVSHGP 的比较，使用了 100 个子模型，即 $M=100$。

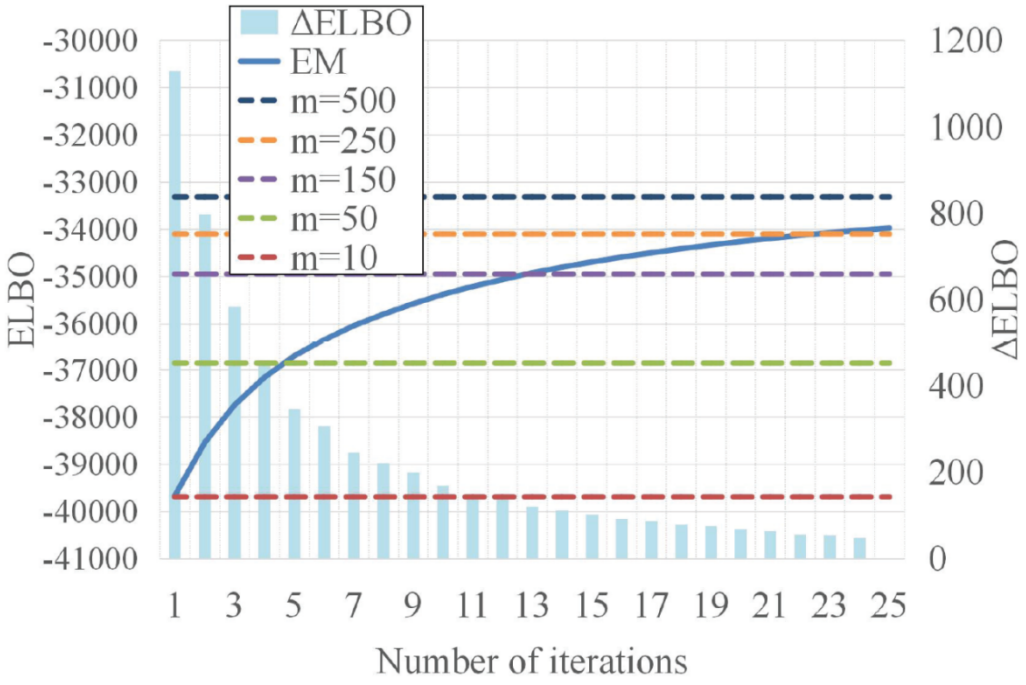

图 5.4 $\text{DVSHGP}_{\text{EM}}$ 模型和有着不同诱导点数量的 DVSHGP 模型在 ELBO 上的比较结果

图 5.4 和 5.5 则说明了 EM 算法在真实数据集也是有效的，其中本文设置了 $m=u=\Delta m=\Delta u=10$。图 5.4 显示了 EM 算法能够通过添加诱导点的方式基本还原直接训练 DVSHGP 模型的 ELBO。从图 5.5 可以看出，当诱导点数量从 10 上升到 150 时，模型有了显著的提升，并且时间花费是较小的。而当诱导点数量从 150 上升到 250 时，则需要有所考虑，因为模型虽然都在 SMSE 与 MSLL 上有所提升，但是增加的时间是非线性的。当诱导点数量从 250 提升到 500 时是不太值

得的，因为此时过多的诱导点使得模型更加关注噪声，而忽略了真实值函数的拟合。

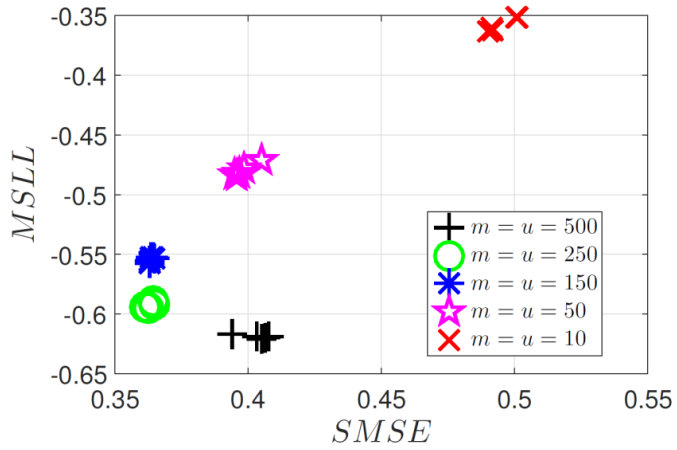

图 5.5　DVSHGP$_{EM}$模型和有着不同诱导点数量的 DVSHGP 模型在预测精度上的比较结果

因此，选择一个合适的早停标准可以平衡模型的花费与精度，虽然如何选择这个标准是灵活的。表 5.1 显示，当选择$\Delta=\psi-2$时，可以很好的平衡时间与精度上的问题。本节之后部分则使用该标准进行应用。比较添加诱导点的策略，是否必须使用重新初始化步骤需要进一步研究，因为重新初始化很有可能导致模型在训练的过程中关于 GP 的超参数得不到充分的训练。图 5.6 则显示了使用重新初始化步骤会使得 EM 算法更加稳定。

表 5.1　在蛋白质数据集上，不同停止标准对 DVSHGP$_{EM}$结果的影响

Δ	t（总迭代次数）	SMSE	MSLL	时间（秒）	ELBO
$\psi-1$	3	0.4395	-0.4200	290.1	-38199.20
$\psi-2$	18	0.3607	-0.5728	5375.1	-34456.02
$\psi-3$	50	0.4010	-0.6195	71726.0	-33209.95

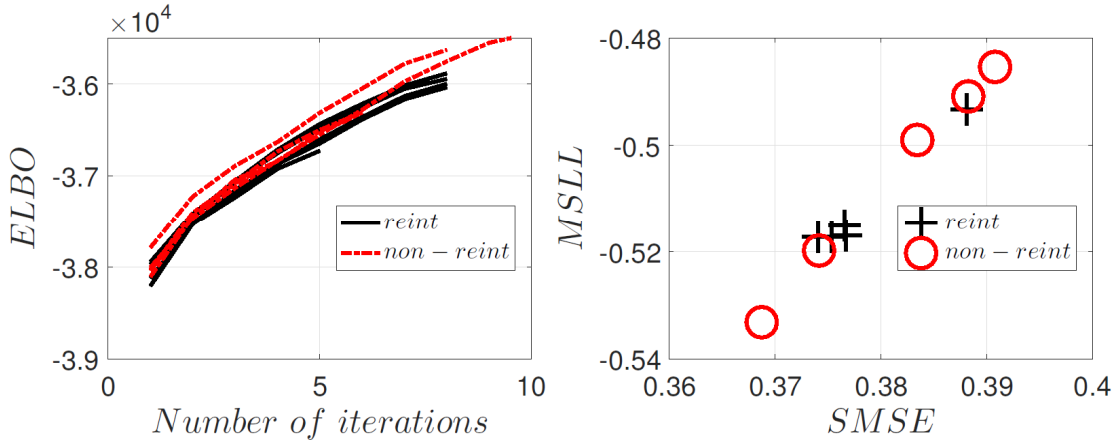

图 5.6　是否使用初始化 GP 超参数步骤对 DVSHGP$_{EM}$在蛋白质数据集上应用的影响

最后，我们考虑对于真实值函数与噪声有着不同数量诱导点的 DVSHGP 模型，结果由图 5.7 表示。将 f 和 g 的部分分开，可以很明显的看出蛋白质数据集的特点，即只需少量的诱导点如 u=50 去模拟变化不大的噪声，而需要大量的诱导点去捕捉快速变化的真实值函数。这同样暗示着，分别考虑 f 和 g 的诱导点可以加快迭代与推断，如对该数据集使用设置 m=u=50、$\Delta m=10$、$\Delta u=0$。注意 $\Delta u=0$ 的结果与 $\Delta m=10$、$\Delta u=10$ 的结果不同主要原因是限制了 $\Delta=\psi-2$，因为两者在每次迭代时添加的诱导点数量不同。

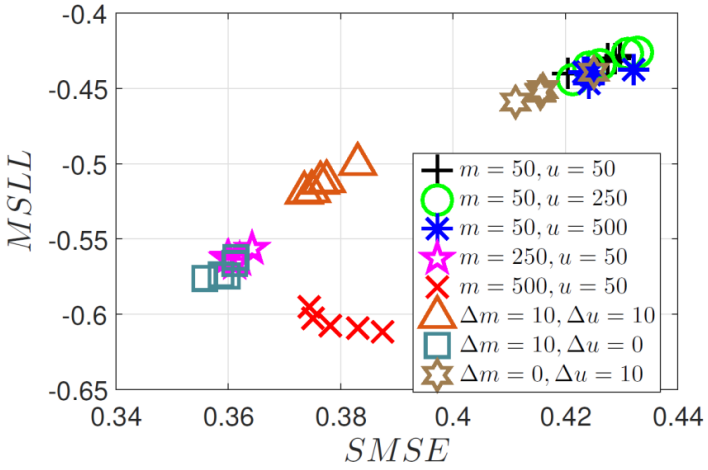

图 5.7　蛋白质数据集上的结果，DVSHGP 模型使用了不同数量的诱导点有 $m\neq u$，$\text{DVSHGP}_{\text{EM}}$ 模型在每次迭代时对于 f 和 g 添加了不同数量的诱导点有 $\Delta m\neq\Delta u$。

5.4.3 现实数据集

本节主要通过四个不同特点的现实数据集研究了算法 5.1 的一些性质，其中早停标准使用 $\Delta=\psi-3$。5 维的翼型数据集 Airfoil[180]，其输出是按比例缩放的声压级，涉及不同风洞速度和迎角下的各种尺寸的 NACA0012 翼型，具有 1503 个数据点[28]。第二个是具有 9568 个实例的 4 维 CCPP 数据集[181][182]，这些数据集是从联合循环发电厂收集了 6 年，以预测每小时的净电能输出。GTCO[183]作为第三个数据集，具有 11 个传感器测量值的 36733 个数据点，这些数据点来自燃气轮机，用于调查烟气排放的 CO。最后的 21 维数据集 SARCOS[9]与机械臂的逆动力学问题相关，它有 44,484 个训练点和 4,449 个测试点。其中 Airfoil 数据集和 CCPP 数据集按 8:2 随机划分训练集与测试集，即分别有 1203/300 和 7655/1913 个样本。对于 GTCO，本文遵循用户协议，使用前 22191 个样本作为训练集，后 14542 个样本作为测试集。作为比较，本文使用了不同的聚合 GP 作为比较——GPoE、RBCM、GRBCM、GPoGRBCM，其中 CGD 的迭代上限设为 $\varsigma=25$。为了表现聚合 GP 的能力，GPoE 和 GPoGRBCM 的第一层使用随机分区，RBCM、

[28] 因为本章的模型不仅考虑了大数据，还考虑了异方差，所以在非大型数据集上测试是必要的。

GRBCM 和 GPoGRBCM 的第二层使用聚类算法分区。每个模型在每个数据集上分别运行 5 次，其结果如图 5.8 所示。

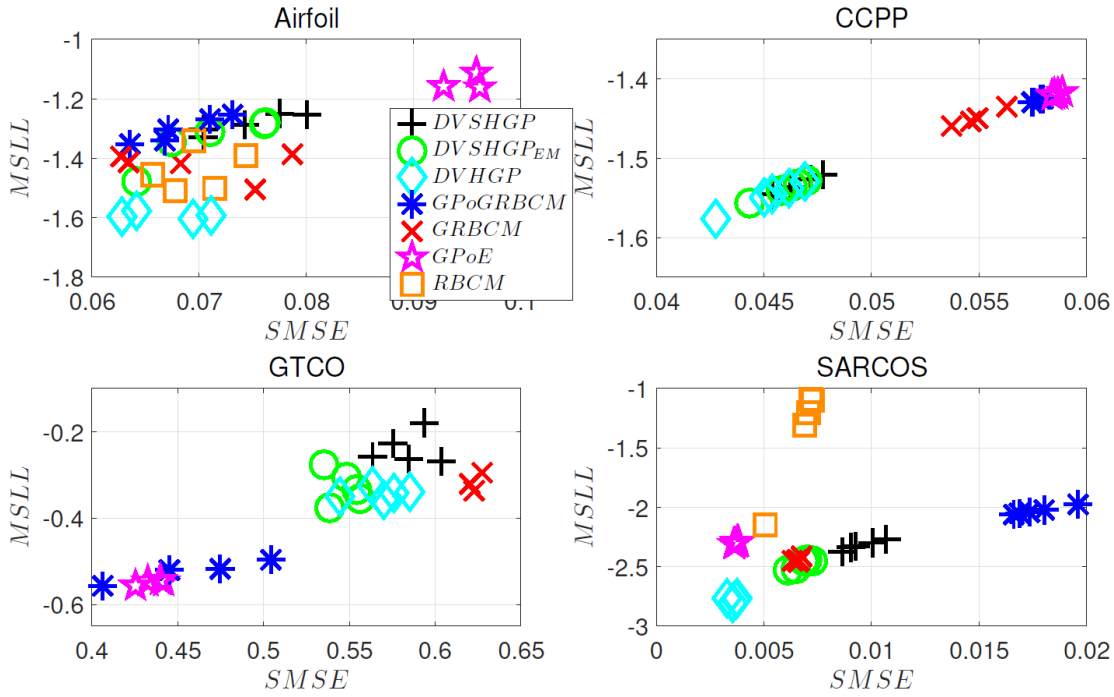

图 5.8 $DVSHGP_{EM}$ 与其他模型在四个真实数据集上的对比结果

（1）Airfoil 数据集：在对比中，为 GPoGRBCM 设置 $M=2\times2$，为其他模型设置 $M=5$，即每个数据子集约有样本 $n_0\approx240$，为 DVSHGP 设置 $m=u=50$ 且 CGD 最大迭代次数为 $\varsigma=100$。对于 DVSHGP_EM 模型，我们设置初始化 $m=u=\Delta m=\Delta u=1$，并且 EM 的迭代上限为 T=150 次。图 5.8 则显示了添加诱导点的方法以及停止标准的有效性。在 EM 算法停止时，DVSHGP_EM 的每个子模型约拥有 77 个诱导点，对比模型可以看出，诱导点的数量从 50 上升到 77 时，无论在 SMSE 还是 MSLL 上都有提升，而从 77 上升到 240 时，模型的预测能力提升不大。但整体上而言，DVHGP 仅花费了 $32.5\pm5.2s$ 得到了几乎最优的结果，一般不需要使用 $DVSHGP_{EM}$。

（2）CCPP 数据集：在对比中，为 GPoGRBCM 设置为 $M=5\times5$，为其他模型设置 $M=25$，即每个数据子集约有样本 $n_0\approx306$，为 DVSHGP 设置 $m=u=50$ 且 CGD 最大迭代次数为 $\varsigma=80$。对于 DVSHGP_EM 模型，我们设置初始化 $m=u=50$、$\Delta m=\Delta u=10$，并且 EM 的迭代上限为 T=25 次。注 DVSHGP 花费 $30.9\pm1.4s$，DVSHGP_EM 花费 $53.8\pm19.4s$，DVHGP 花费 $193.4\pm32.5s$。这说明了，当初始化的诱导点个数已经足够总结数据集的时候，EM 算法并不会使时间花费多出非常多，而是在可接受的范围。

（3）GTCO 数据集：在对比中，为 GPoGRBCM 设置为 $M=7\times10$，为其他模型设置 $M=70$，即每个数据子集约有样本 $n_0\approx317$，为 DVSHGP 设置

$m=u=50$且 CGD 最大迭代次数为$\varsigma=30$。对于 DVSHGP_EM 模型，我们设置初始化$m=u=50$、$\Delta m=\Delta u=10$，并且 EM 的迭代上限为 T=10 次。图 5.8 说明 DVSHGP_EM 花费$45.6\pm2.9s$取得了 HGP 中最优的结果，而不使用稀疏近似的 DVHGP 则花费了$359.9\pm88.9s$。此外，GPoE 和 GPoGRBCM 在这个数据集中取得了优秀的结果，说明了这个数据集本身的异方差性不显著，使用聚类分区的其他模型过度注重了局部的趋势。

（4）SARCOS 数据集：在对比中，为 GPoGRBCM 设置为$M=10\times15$，为其他模型设置$M=150$，即每个数据子集约有样本$n_0\approx297$，为 DVSHGP 设置$m=u=80$且 CGD 最大迭代次数为$\varsigma=30$。对于 DVSHGP_EM 模型，我们设置初始化$m=u=80$、$\Delta m=\Delta u=20$，并且 EM 的迭代上限为 T=10 次。图 5.8 暗示了 DVSHGP_EM 可以作为 DVHGP 的优良替代品，其中 DVSHGP 花费$81.2\pm1.7s$，DVSHGP_EM 花费$221.1\pm91.7s$，DVHGP 则花费$1131.4\pm124.4s$。对于该数据集，GPoE 与 GRBCM 表现相对不错，说明数据集的异方差性较弱。而 DVSHGP 超出 DVSHGP_EM 比较大一部分，说明在添加诱导点的时候，可以考虑多拟合真实值函数 f，如使用$\Delta m=30$和$\Delta u=10$。

5.5 结论与讨论

为了探寻对于 HGP 而言，需要多少个诱导点这个问题，本文主要基于(D)VSHGP 的 ELBO，拓展了先验策略与后验策略以供诱导点数量的选择。先验策略则提供一个清晰的诱导点数量 m 关于样本数量 n 的关系。后验策略则通过 EM 算法一步一步迭代，以不断添加诱导点的方式以至期望精度。然而，两种策略都存在一些潜在的问题。对于先验策略，在训练之前很难得到正确的核函数参数，以至于难以得到精确的诱导点数量的点估计。对于后验策略，如果我们将 m、u、Δm、Δu 设置得过小，在数据特征变化非常快速的情况，难以完成局部最优到全局最优的跨越。另外，当数据集具有大数据、高维等特征时，重新初始化的步骤也在实际应用中存在局限性，因为此时每次迭代中训练超参数时需要花费大量的步数，而步数不足时则容易出现本次结果不如前一次的情况。

6 讨论与展望

MacKay（1998）[184]曾经提问道：“高斯过程怎样才有可能取代神经网络？我们把婴儿连同洗澡水一起倒掉了吗？[29]”当然，这是在当时的背景下，研究人员对于神经网络的使用还未构建出原则性的框架，如：如何设计网络结构、如何选择激活函数等。如今，也有众多工作解决上述疑惑，如 Auto Machine Learning。但回顾该问题总归是有好处的，并且伴随着体会到神经网络在工业界被广泛应用，而高斯过程模型相对地仅在学术界被广泛地关注，思考高斯过程到底能为神经网络带来什么样的好处？现已知广泛的神经网络结构已经能满足多种工业场景的需求了。或者，若是提供预测的不确定性，则贝叶斯网络也可以满足需求；若是需要通过有表现力的核函数来自主识别数据的结构特征，使用带正则项的回归模型是否可行；使用其他的随机过程替代除了在推断方法是否有其他的影响？当然这是在结合高斯过程与神经网络的基础上提问的，并没有其中一个模型被抛弃，一个典型的想法便是结合两者，并扬长避短。并且无法否认的是，将两者结合依旧是一块活跃的研究领域。

那么如何改进模型呢？对于模型的研究，一方面可以从理解模型入手，如上海交通大学的许志钦老师尝试理解神经网络是如何学习的，得到了学习的“频率准则”。该方面所需的实践相对较少，所需的反馈从数值实验即可得到结果。另一方面，可以从模型的改进方向入手，如处理计算机视觉方面的问题而导出卷积神经网络、解决自然语言处理方面导出循环神经网络。对于第一方面，本人当前的目光仅局限于如果将高斯过程回归模型应用于大数据，即对模型本身的表现能力无任何影响，而对于模型的改进认知，也仅局限于非平稳性、异方差性，或是初始超参数的影响[185]，以及核函数的选取[186]等，与神经网络模型的改进相比似乎是微不足道的。当然，该方面认知的局限性可能是由数学水平不足造成的，若学习更多随机过程的性质，并使理解上升到泛函层面，可能会有新的研究方向。对于第二方面，理论方面的创新是优美的，但将目光局限于理论同样会抑制理论创新，反而有时从应用中得到的反馈更具启发意义，如一些启发式算法，以及洛伦兹发现混沌的经历，所以需将模型与应用领域或者目标结合。

神经网络本身即是模拟神经系统的产物，但其还原神经系统的能力还稍显不足，那么是否可将高斯过程与神经网络结合，用来描述神经系统呢？一个简单但

[29] 原文为：How can Gaussian processes possibly replace neural networks? Have we thrown the baby out with the bathwater?

新颖的结合方法可以参考神经元高斯过程[43]。但是，本人并未清楚赋予机器学习模型生物学习能力的优越性何在，如脉冲神经网络，也不清楚使用神经元高斯过程是否足够表达神经元乃至神经系统的复杂性。在此之上，本人也未理清复杂系统的分析范式，即如何将系统还原到小单位局部分析，而在整体分析时附加非线性影响，同时也未具备展望人工智能发展方向的能力。但这些问题的答案，作者期望能在将高斯过程与神经网络的结合体应用于神经系统领域的过程中得到。

对于复杂系统的描述，鉴于 2021 年诺贝尔物理学奖表彰“理解复杂物理系统”方面的贡献，物理领域的研究方法值得借鉴。对于人工智能的展望，可从需求端着手，即需要满足怎样的内容，然后层层递推回如何改进模型。但从推崇非监督学习而不是监督学习来表现智能的观察来看，局限于现有统计模型框架研究可能并不能满足“智能”的需求。

参考文献

[1] 李金昌. 话说“回归”[J]. 中国统计，2020，68(10)：31-33.

[2] （奥）薛定谔. 生命是什么[M]. 海南：海南出版社，2017：1-97.

[3] （比）普里戈金. 从混沌到有序[M]. 上海：上海译文出版社，2005：1-314.

[4] 尼克. 人工智能简史[M]. 北京：人民邮电出版社，2017：1-255.

[5] Murphy, K.P.. Machine Learning: A Probabilistic Perspective[M]. Massachusetts: The MIT Press, 2012: 1-24.

[6] Bishop, C.M.. Pattern Recognition and Machine Learning[M]. New York: Springer Science+Business Media, 2007: 1-55.

[7] Hastie, T., Tibshirani, R., Wainwright, M.. Statistical Learning with Sparsity: The Lasso and Generalizations[M]. Los Angeles: CRC Press, 2015: 1-7.

[8] Shalev-Shwartz, S., Ben-David, S.. Understanding Machine Learning: From Theory to Algorithms[M]. Cambridge: Cambridge University Press, 2014: 60-66.

[9] Rasmussen, C.E., Williams, C.K.I.. Gaussian Processes for Machine Learning[M]. Massachusetts: The MIT Press, 2006: 1-218.

[10] Le, N.D., Zidek, J.V.. Statistical Analysis of Environmental Space-Time Processes[M]. New York: Springer Science+Business Media, 2006: 83-116.

[11] Cressie, N.. The origins of kriging[J]. Mathematical Geology, 1990, 22(03): 239-252.

[12] Chiles, J., Delfiner, P.. Geostatistics: Modeling Spatial Uncertainty (Second Edition)[M]. New Jersey: John Wiley & Sons, 2012: 31-238.

[13] Wahba, G.. Spline Models for Observational Data[M]. Pennsylvania: Society for Industrial and Applied Mathematics, 1990: 1-45.

[14] 牛文杰. 薄板样条法和泛克里金法在理论和应用方面的比较[J]. 工程图学学报, 2010, 31(04): 123-129.

[15] Bobrowski, A.. Functional Analysis for Probability and Stochastic Processes: An Introduction[M]. Cambridge: Cambridge University Press, 2005: 1-101.

[16] Brockwell, P.J., Davis, R.A.. Time Series: Theory and Methods (Second Edition)[M]. New York: Springer Science+Business Media, 2006: 1-76.

[17] Williams, D.. Probability with Martingales[M]. Cambridge: Cambridge University Press, 1991: 83-92.

[18] Banerjee, A., Guo X., Wang H.. On the optimality of conditional expectation as a Bregman predictor[J]. IEEE Transactions on Information Theory, 2005, 51(07): 2664-2669.

[19] Forrester, A.I.J., Sobester, A., Keane, A.J.. Engineering Design via Surrogate Modelling: A Practical Guide[M]. West Sussex: John Wiley & Sons, 2008: 33-153.

[20] Vazquez, E., Bect, J.. Pointwise consistency of the kriging predictor with known mean and covariance functions[A]. Giovagnoli A., Atkinson A., Torsney B., May C. (eds) mODa 9 – Advances in Model-Oriented Design and Analysis[C]. Berlin: Physica-Verlag HD, 2009: 221-228.

[21] Choi, T., Schervish, M.J.. Posterior consistency in nonparametric regression problems under Gaussian process priors[R]. Carnegie Mellon University, 2004.

[22] Neal, R.M.. Bayesian Learning for Neural Networks[D]. Ph.d. thesis, University of Toronto, 1995.

[23] Hanin, B.. Random neural networks in the infinite width limit as Gaussian processes[J]. arXiv preprint arXiv: 2107.01562, 2021.

[24] Lee, J., Bahri, Y., Novak, R., Schoenholz, S.S., Pennington, J., Sohl-Dickstein, J.. Deep neural networks as Gaussian processes[A]. Sixth International Conference on Learning Representations[C]. Vancouver: ICLR, 2018: 1-17.

[25] Williams, C.K.I.. Computing with infinite networks[A]. Advances in neural information processing systems 10[C]. Colorado: NeurIPS, 1997: 295-301.

[26] Cho, Y., Saul, L.K.. Kernel methods for deep learning[A]. Advances in Neural Information Processing Systems 22[C], Massachusetts: the MIT Press, 2009: 342-350.

[27] Matthews, A.G.D.G., Hron, J., Rowland, M., Turner, R.E., Ghahramani, Z.. Gaussian Process Behaviour in Wide Deep Neural Networks[A]. Sixth International Conference on Learning Representations[C]. Vancouver: ICLR, 2018: 1-15.

[28] Lee, C., Wu, J., Wang, W., Yue, X.. Neural Network Gaussian Process considering Input Uncertainty for Composite Structures Assembly[J]. IEEE/ASME Transactions on Mechatronics, 2020, doi: 10.1109/TMECH.2020.3040755.

[29] Pretorius, A., Kamper, H., Kroon, S.. On the expected behaviour of noise regularised deep neural networks as Gaussian processes[J]. Pattern Recognition

Letters, 2020, 138: 75-81.

[30] Garriga-Alonso, A., Rasmussen, C.E., Aitchison, L.. Deep convolutional networks as shallow Gaussian processes[A]. Seventh International Conference on Learning Representations[C]. Louisiana: ICLR, 2019: 1-16.

[31] Novak, R., Xiao, L., Lee, J., Bahri, Y., Yang, G., Abolafia, D.A., Pennington, J., Sohl-Dickstein, J.. Bayesian deep convolutional networks with many channels are Gaussian processes[A]. Third workshop on Bayesian Deep Learning (NeurIPS 2018)[C]. Canada: NeurIPS, 2018, 1-27.

[32] Wilson, A.G., Hu, Z., Salakhutdinov, R., Xing, E.P.. Deep kernel learning[A]. 19th International Conference on Artificial Intelligence and Statistics[C]. PRML: W&CP, 2016, 51: 370-378.

[33] Wilson, A.G., Hu, Z., Salakhutdinov, R., Xing, E.P.. Stochastic variational deep kernel learning[A]. 30th International Conference on Neural Information Processing Systems[C]. New York: Curran Associates Inc., 2016, 2594-2602.

[34] Bui, T.D., Hernandez-Lobato, D.H., Li, Y., Hernandez-Lobato, J.M.. Deep Gaussian processes for regression using approximate expectation propagation[J]. arXiv preprint arXiv: 1602.04133, 2016.

[35] Damianou, A., Lawrence, N.D.. Deep Gaussian processes[A]. Sixteenth International Conference on Artificial Intelligence and Statistics[C]. PMLR, 2013, 31: 207-215.

[36] Jain, A., Srijith, P.K., Khan, M.E.. Subset-of-data variational inference for deep Gaussian-processes regression[A]. 37th Conference on Uncertainty in Artificial Intelligence[C]. PMLR, 2021, (in press): 1-15.

[37] Pleiss, G., Cunningham, J.P.. The limitations of large width in neural networks: a deep Gaussian process perspective [J]. arXiv preprint arXiv: 2106.06529, 2021.

[38] Duvenaud, D., Rippel, O., Adams, R.P., Ghahramani, Z.. Avoiding pathologies in very deep networks[A]. 17th International Conference on Artificial Intelligence and Statistics[C]. PRML: W&CP, 2016, 33: 202-210.

[39] Dunlop, M.M., Girolami, M.A., Stuart, A.M., Teckentrup, A.L.. How Deep Are Deep Gaussian Processes?[J]. Journal of Machine Learning Research, 2018, 19(54): 1-46.

[40] Garnelo, M., Rosenbaum, D., Maddison C.J., Ramalho, T., Saxton, D., Shanahan, M., Teh, Y.W., Rezende D.J., Eslami S.M.A.. Conditional neural processes[A]. 35th International Conference on Machine Learning[C]. PMLR, 2018, 80:

1704-1713.

[41] Garnelo, M., Schwarz, J., Rosenbaum, D., Viola, F., Rezende D.J., Eslami S.M.A., Teh, Y.W.. Neural Processes[A]. the ICML 2018 workshop on Theoretical Foundations and Applications of Deep Generative Models[C]. ICML, 2018, 1-11.

[42] Kim, H., Mnih, A., Schwarz, J., Garnelo, M., Eslami, A., Rosenbaum, D., Vinyals, O., Teh, Y.W.. Attentive neural processes[A]. Third workshop on Bayesian Deep Learning (NeurIPS 2018)[C]. NeurIPS, 2018, 1-17.

[43] Friedrich, J.. Neuronal Gaussian process regression[A]. Advances in Neural Information Processing Systems 34[C]. Vancouver: NeurIPS, 2020, (in press): 1-11.

[44] Hastie, T., Tibshirani, R., Friedman, J.. The Elements of Statistical Learning: Data Mining, Inference, and Prediction (Second Edition)[M]. Springer, 2017, 139-189.

[45] Raket, L.L.. Differential equations, splines and Gaussian processes[J]. arXiv preprint arXiv: 2102.03306, 2021.

[46] Alvarez, M., Luengo, D., Lawrence, N.D.. Latent force model[A]. 12th International Conference on Artificial Intelligence and Statistics (AISTATS)[C]. JMLR: W&CP, 2009, 5: 9-16.

[47] McDonald, T.M., Alvarez, M.A.. Computational modeling of nonlinear dynamical systems with ODE-based random features[J]. arXiv preprint arXiv: 2106.0596, 2021.

[48] Lindgren, F., Rue, H., Lindstrom, J.. An explict link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differenial equation approach[J]. Journal of the Royal Statistical Society: Series B, 2011, 73(04): 423-498.

[49] Rasmussen, C.E.. Evaluation of Gaussian processes and other methods for non-linear regression[D]. Ph.d. thesis, University of Toronto, 1999.

[50] Liu, H., Cai, J., Ong, Y., Wang, Y.. Understanding and comparing scalable Gaussian process regression for big data[J]. Knowledge-Based Systems, 2019, 164:324-335.

[51] Quinonero-Candela, J., Rasmussen, C.E.. A Unifying View of Sparse Approximate Gaussian Process Regression[J]. Journal of Machine Learning Research, 2005, 6: 1939-1959.

[52] Williams, C.K.I., Seeger, M.. Using the Nystrom Method to Speed Up Kernel Machines[A]. Advances in Neural Information Processing Systems 13[C]. MIT Press, 2001.

[53] Snelson, E., Ghahramani, Z.. Local and global sparse Gaussian process approximations[A]. Eleventh International Conference on Artificial Intelligence and Statistics[C]. PMLR, 2: 524-531.

[54] Low, K.H., Yu, J., Chen, J., Jaillet, P.. Parallel Gaussian process regression for big data: low-rank representation meets Markov approximation[A]. Association for the Advancement of Artificial Intelligence 2015[C]. AAAI Press, 2015, 1-10.

[55] Snelson, E., Ghahramani, Z.. Sparse Gaussian Process Using Pseudo-inputs[A]. Advances in Neural Information Processing Systems 18[C], MIT Press, 2006, 18: 1257--1264.

[56] Walder, C., Kim, K.I., Scholkopf, B.. Sparse multiscale Gaussian process regression[A]. 25th International Conference on Machine Learning[C]. ICML, 2008, 1112-1119.

[57] Bui, T., Turner R.. Tree-structured Gaussian Process Approximations[A]. Advances in Neural Information Processing Systems 26[C]. NeurIPS, 2014, 3: 2213-2221.

[58] Figueirasvidal, A., Lázarogredilla M.. Inter-domain Gaussian Processes for Sparse Inference using Inducing Features[A]. Advances in Neural Information Processing Systems 21[C], MIT Press, 2009, 21: 1087--1095.

[59] Lázaro-Gredilla, M., Quiñonero-Candela, J., Rasmussen, C.E., Figueiras-Vidal A.R.. Sparse Spectrum Gaussian Process Regression[J]. Journal of Machine Learning Research, 2010, 11(9):1865-1881.

[60] Hoang, Q.M., Hoang, T.N., Pham, H., Woodruff, D.P.. Revisiting the sample complexity of sparse spectrum approximation of Gaussian processes[A]. 34th Conference on Neural Information Processing Systems[C]. NeurIPS, 2020, 1-11.

[61] Wilson, A.G., Adams, R.P.. Gaussian process kernels for pattern discovery and extrapolation[A]. 30th International Conference on Machine Learning[C]. JMLR: W&CP, 2013, 28: 1067-1075.

[62] Rahimi, A., Recht, B.. Random features for large-scale kernel machines[A]. 20th International Conference on Neural Information Processing Systems[C]. New York: Curran Associates Inc., 2007, 1177-1184.

[63] Solin, A., Särkkä, S. Hilbert space methods for reduced-rank Gaussian process

regression[J]. Statistics and Computing, 2019, 30: 419–446.

[64] Bengio, Y., Vincent, P., Paiement, J.. Spectral clustering and kernel PCA are learning eigenfunctions[R]. CIRANO Working Papers, 2003, 2003s-19.

[65] Wilson, A.G., Dam, C., Nichisch, H.. Thoughts on massively scalable Gaussian processes[J]. arXiv preprint arXiv: 1511.0187, 2015.

[66] Titsias, M.K.. Variational learning of inducing variables in sparse Gaussian processes[A]. 12th International Conference on Artificial Intelligence and Statistics[C]. JMLR: W&CP, 2009, 5: 567-574.

[67] Titsias, M.K.. Variational Model Selection for Sparse Gaussian Process Regression[R]. 2009.

[68] Hoang, T.N.. A distributed variational inference framework for unifying parallel sparse Gaussian process regression models[A]. 33rd International Conference on International Conference on Machine Learning[C]. PLMR, 2016, 48: 382-391.

[69] Gal, Y., Mark, V., Rasmussen, C.E.. Distributed Variational Inference in Sparse Gaussian Process Regression and Latent Variable Models[A]. Advances in Neural Information Processing Systems 2014[C]. NeurIPS, 2014, 1-9.

[70] Liu H., Ong Y., Cai, J.. Large-scale heteroscedastic regression via Gaussian process[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32(02): 708-721.

[71] Hensman, J., Fusi, N., Lawrence, N.D.. Gaussian processes for big data[J]. arXiv preprint arXiv: 1309.6835, 2013.

[72] Hoang, T.N., Hoang, Q.M., Low, K.H.. A unifying framework of anytime sparse Gaussian process regression models with stochastic variational inference for big data[A]. 32nd International Conference on International Conference on Machine Learning[C]. ICML, 2015, 37: 569-578.

[73] Bui, T. D., Yan, J., Turner, R.E.. A unifying framework for Gaussian process pseudo-point approximations using power expectation propagation[J]. Journal of Machine Learning Research, 2016, 18: 1-72.

[74] Liu H., Ong, Y., Shen, X., Cai, J.. When Gaussian Process Meets Big Data: A Review of Scalable GPs[J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, 31(11): 4405-4423.

[75] Hernandez-Lobato, D., Hernandez-Lobato, J.M., Li, Y., Bui, T., Turner, R.E.. Stochastic expectation propagation for large scale Gaussian process classification[J]. arXiv preprint arXiv: 1511.03249, 2015.

[76] Matthews, A.G.D.G., Hensman, J., Turner, R.E., Ghahramani, Z.. On sparse variational methods and the Kullback-Leibler divergence between stochastic processes[A]. 19th International Conference on Artificial Intelligence and Statistics[C]. JMLR: W&CP, 2016, 41: 231-239.

[77] Hensman, J., Matthews, A.G.D.G., Filippone, M., Ghahramani, Z.. MCMC for variationally sparse Gaussian processes[A]. Advances in Neural Information Processing Systems 2015[C]. NuerIPS, 2015, 1648-1656.

[78] Huggins, J.H., Campbell, T., Kasprzak, M., Broderick, T.. Scalable Gaussian Process Inference with Finite-data Mean and Variance Guarantees[A]. 22nd International Conference on Artificial Intelligence and Statistics[C]. PMLR, 2019, 89: 1-20.

[79] Hensman, J., Durrande, N., Solin, A.. Variational fourier features for Gaussian processes[J]. The Journal of Machine Learning Research, 2017, 18(01): 5537-5588.

[80] Gal, Y., Turner, R.. Improving the Gaussian process sparse spectrum approximation by representing uncertainty in frequency inputs[A]. 32nd International Conference on International Conference on Machine Learning[C]. JMLR: W&CP, 2015, 37: 655-664.

[81] Bauer, M., Wilk, M.V.D., Rasmussen, C.E.. Understanding probabilistic sparse Gaussian Process approximations. Advances in Neural Information Processing Systems 2016[C]. NeurIPS, 2016, 1533-1541.

[82] Gibbs, M., Mackay, D.. Efficient implementation of Gaussian processes[R]. 1997.

[83] Gray, A.. Fast kernel matrix-vector multiplication with application to Gaussian process learning[R]. Carnegie Mellon University, 2004.

[84] Pleiss, G., Jankowiak, M., Eriksson, D., Damle, A., Gardner, J.R.. Fast matrix square roots with applications to Gaussian processes and Bayesian optimization[A]. 34th Conference on Neural Information Processing Systems[C]. NeurIPS, 2020, 1-14.

[85] Quinonero-Candela, J., Ramussen, C.E., Williams, C.K.I.. Approximation methods for Gaussian process regression[R]. MIT Press, 2007.

[86] Murray, I.. Gaussian processes and fast matrix-vector multiplies[A]. Workshop on numerical mathematics at the 26th International Conference on Machine Learning[C]. ICML, 2009, 1-4.

[87] Vanhatalo, J., Vehtari, A.. Modelling local and global phenomena with sparse Gaussian processes[J]. arXiv preprint arXiv: 1206.3290, 2012.

[88] Wilson, A.G., Nickisch, H.. Kernel interpolation for scalable structured Gaussian processes (KISS-GP)[A]. 32nd International Conference on International Conference on Machine Learning[C]. PRML, 2015, 37: 1775-1784.

[89] Fritz, J., Neuweiler, I., Nowak, W.. Application of FFT-based algorithms for large-scale universal Kriging problems[J]. Mathematical Geosciences, 2009, 41(5): 509-533.

[90] Shen, Y., Ng, A.Y., Seeger, M.. Fast Gaussian process regression using KD-Trees[A]. 18th International Conference on Neural Information Processing Systems[C]. NeurIPS, 2005, 1225-1232.

[91] Wang, K.A., Pleiss, G., Gardner, J.R., Tyree, S., Weinberger, K.Q., Wilson, A.G.. Exact Gaussian processes on a million data points[A]. 33rd Conference on Neural Information Processing Systems[C]. NeurIPS, 2019, 1-13.

[92] Yang, C., Duraiswami, R., Davis, L.. Efficient kernel machines using the improved fast Gauss transform[A]. Advances in neural information processing systems17[C] NeurIPS, 2004, 1561-1568.

[93] Chen J., Wang, L., Anitescu, M.. A fast summation tree code for Matern kernel[J]. SIAM Journal On Scientific Computing, 2014, 36(01): 289-309.

[94] Pleiss, G., Grandner, J.R., Weinberger, K.Q., Wilson, A.G.. Constant-time predictive distributions for Gaussian processes[A]. 35th International Conference on Machine Learning[C]. PMLR, 2018, 80: 4114-4123.

[95] Chalupka, K., Williams, C.K.I.. A framework for evaluating approximation methods for Gaussian process regression[J]. arXiv preprint arXiv: 1205.6326, 2012.

[96] Nguyen, D., Filippone, M., Michiardi, P.. Exact Gaussian process regression with distributed computations[A]. 34th ACM/SIGAPP Symposium on Applied Computing[C]. New York: Association for Computing Machinery, 2019, 1286-1295.

[97] Dietrich, C.R., Newsam, G.N.. Fast and exact simulation of stationary Gaussian processes through circulant embedding of the covariance matrix[J]. SIAM Journal on Scientific Computing, 1997, 18(4): 1088-1107.

[98] Ambikasaran, S., Foreman-Mackey, D., Greengard, L., Hogg D.W. O'Neil, M.. Fast direct methods for Gaussian processes[J]. IEEE Transactions on Pattern

Analysis and Machine Intelligence, 2016, 38(02): 252-265.

[99] Xu, Y., Yin, F., Zhang, J., Xu, W., Cui, S., Luo, Z.. Scalable Gaussian process using inexact admm for big data[A] 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)[C]. IEEE, 2019, 1-5.

[100] Chen, H., Zheng, L., Kontar, R.A., Raskutti., G.. Stochastic gradient descent in correlated settings: a study on Gaussian processes[A]. 34th Conference on Neural Information Processing Systems[C]. NeurIPS, 2020, 1-12.

[101] Almosallam, I.A., Jarvis, M.J., Roberts, S.J.. GPz: non-stationary sparse Gaussian processes for heteroscedastic uncertainty estimation in photometric redshifts[J]. Monthly Notices of the Royal Astronomical Society, 2016, 462(1): 726-739.

[102] Raissi, M.. Parametric Gaussian process regression for big data[J]. arXiv preprint arXiv: 1704.03144, 2017.

[103] Sarkka, S., Solin, A., Hartikainen J.. Spatiotemporal Learning via Infinite-Dimensional Bayesian Filtering and Smoothing: A Look at Gaussian Process Regression Through Kalman Filtering[J]. IEEE Signal Processing Magazine, 2013, 30(4): 51-61.

[104] Hartikainen, J., Särkkä, S.. Kalman filtering and smoothing solutions to temporal Gaussian process regression models[A]. 2010 IEEE International Workshop on Machine Learning for Signal Processing[C]. IEEE, 2010, 379-384.

[105] Samo, Y.K., Roberts, S.. String and membrane Gaussian processes[J]. The Journal of Machine Learning Research, 2016, 17(01): 4485-4571.

[106] Choudhury, A., Nair, P.B., Keane, A.J.. A data parallel approach for large-scale Gaussian process modeling[A]. 2020 Society for Industrial & Applied Mathematics (SIAM) International Conference on Data Mining[C]. SDM, 2002, 95–111.

[107] Gramacy, R.B., Lee, H.K.. Bayesian Treed Gaussian Process Models With an Application to Computer Modeling[J]. Journal of the American Statistical Association, 2008, 103(483): 1119-1130.

[108] Park, C., Huang, J.Z., Ding, Y.. Domain Decomposition Approach for Fast Gaussian Process Regression of Large Spatial Data Sets[J]. Journal of Machine Learning Research, 2011, 12(4): 1697−1728.

[109] Park, C., Huang, J.. Efficient Computation of Gaussian Process Regression for Large Spatial Data Sets by Patching Local Gaussian Processes[J]. Journal of

Machine Learning Research, 2016, 17(1): 6071-6099.

[110] Park, C., Apley, D.. Patchwork Kriging for large-scale Gaussian process regression[J]. Journal of Machine Learning Research, 2018, 19(1): 269-311.

[111] Urtasun, R., Darrell, T.. Sparse probabilistic regression for activity-independent human pose inference[A]. 2008 IEEE Conference on Computer Vision and Pattern Recognition[C]. IEEE, 2008, 1-8.

[112] Moore, D., Russell, S.J.. Gaussian Process Random Fields[A]. Advances in Neural Information Processing Systems 28[C]. NeurIPS, 2015, 21(3): 763-772.

[113] Das, S., Roy, S., Sambasivan, R.. Fast Gaussian process regression for big data[J]. Big Data Research, 2018, 14: 12-26.

[114] Chen, T., Ren, J.. Bagging for Gaussian process regression[J]. Neurocomputing, 2009, 72(7-9): 1605-1610.

[115] Zhu, J., Jiang, M., Peng, G., Yao, L., Ge, Z.. Scalable soft sensor for nonlinear industrial big data via bagging stochastic variational Gaussian processes[J]. IEEE Transactions on Industrial Electronics, 2021,68(08): 7594-7602.

[116] Cao, Y., Fleet, D.J.. Generalized product of experts for automatic and principled fusion of Gaussian process predictions[J]. arXiv preprint arXiv: 1410.7827, 2015.

[117] Hinton, G.E.. Training products of experts by minimizing contrastive divergence[J]. Neural Computation, 2002, 14(08): 1771-1800.

[118] Deisenroth, M.P., Ng, J.W.. Distributed Gaussian processes[A]. 32nd International Conference on Machine Learning[C]. PMLR: W&CP, 2015, 37: 1481-1490.

[119] Tresp, V.. A Bayesian committee machine[J]. Neural computation, 2000, 12(11): 2719-2741.

[120] Liu, H., Cai, J., Wang, Y., Ong, Y.S.. Generalized robust bayesian committee machine for large-scale Gaussian process regression[A]. 35th International Conference on Machine Learning[C]. PMLR, 2018, 80: 3131-3140.

[121] Li, N., Gao, Y., Li, W., Jiang Y., Xia S.. H-GPR: a hybrid strategy for large-scale Gaussian process regression[A]. 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)[C]. IEEE, 2021, 2955-2959.

[122] Rullière, D., Durrande, N., Bachoc, F. Chevalier, C.. Nested Kriging predictions for datasets with a large number of observations[J]. Statistics and Computing, 2018, 28: 849–867.

[123] Bachoc, F., Durrande, N., Rullière, D., Chevalier, C.. Some properties of nested Kriging predictors[J]. arXiv preprint arXiv: 1707.05708, 2017.

[124] Ng, J.W., Deisenroth, M.P.. Hierarchical mixture-of-experts model for large-scale Gaussian process regression[J]. arXiv preprint arXiv: 1412.3078, 2014.

[125] Da, B., Ong, Y., Gupta, A., Feng, L., Liu, H.. Fast transfer Gaussian process regression with large-scale sources[J]. Knowledge-Based Systems, 2019, 165: 208-218.

[126] Gao, Y., Li, N., Ding, N., Li, Y., Dai, T., Xia, S.. Generalized local aggregation for large scale Gaussian process regression[A]. 2020 International Joint Conference on Neural Networks (IJCNN)[C] IEEE, 2020, 1-8.

[127] 刘晓芳, 刘策, 刘露咪, 程丹松. 重叠局部高斯过程回归[J]. 哈尔滨工业大学学报, 2019, 51(11): 22-26.

[128] Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.. Adaptive mixture of local experts[J]. Neural computation, 1991, 3: 79-87.

[129] Rasmussen, C.E., Ghahramani, Z.. Infinite mixtures of Gaussian process experts[A]. 14th International Conference on Neural Information Processing Systems: Natural and Synthetic[C]. NeurIPS, 2001, 881-888.

[130] Meeds, E., Osindero, S.. An alternative infinite mixture of Gaussian process experts[A] Advances in Neural Information Processing Systems 18[C]. NeuraIPS, 2005, 1-8.

[131] Nguyen, T.N.A., Bouzerdoum, A., Phung, S.L. Variational inference for infinite mixtures of sparse Gaussian processes through KL-correction[A]. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)[C]. IEEE, 2016, 2579-2583.

[132] Nguyen, T.V., Bonilla, E.V.. Fast allocation of Gaussian process experts[A]. 31st International Conference on International Conference on Machine Learning[C]. JMLR, 2014, 32: 145-153.

[133] Shi, J.Q., Murray-Smith, R., Titterington, D.M.. Hierarchical Gaussian process mixtures for regression[J]. Statistics and computing, 2005, 15(1): 31-41.

[134] Nguyen-Tuong, D., Seeger, M., Peters, J., Koller, D., Bottou, L.. Local Gaussian process regression for real time online model learning and control[A]. Advances in Neural Information Processing Systems 21[C]. NeurIPS, 2008, 1193-1200.

[135] Nguyen, T.N.A., Bouserdoum, A., Phung, S.L.. Scalable hierarchical mixture of Gaussian processes for pattern classification[A]. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)[C]. IEEE, 2018, 2466-2470.

[136] Nguyen, T.N., Bouzerdoum, A., Phung, S.L.. Stochastic variational hierarchical mixture of sparse Gaussian processes for regression[J]. Machine Learning, 2018, 107(12): 1947-1986.

[137] Trecate, G.F., Williams, C.K.I., Opper, M.. Finite-dimensional approximation of Gaussian processes[A]. Advances in Neural Information Processing Systems 11[C]. MIT Press, 1999, 218-224.

[138] Zhu, H., Williams, C.K.I., Rohwer, R., Morciniec, M.. Gaussian regression and optimal finite dimensional linear models[J]. Neural networks and machine learning, 1997, 167-184.

[139] Dereziński, M., Khanna R., Mahoney, M.W.. Improved guarantees and a multiple-descent curve for the column subset selection problem and the Nystrom method[A]. Advances in Neural Information Processing Systems 34[C]. Vancouver: NeurIPS, 2020, 1-12.

[140] Burt, D.R., Rasmussen, C.E., Wilk, M.V.D.. Rates of convergence for sparse variational Gaussian process regression[A]. 36th International Conference on Machine Learning[C]. PMLR, 2019, 97: 862-871.

[141] Raykar, V.C., Duraiswami, R.. Fast large scale Gaussian process regression using approximate matrix-vector products[R]. 2007.

[142] Smola, A., Bartlett, P.. Sparse Greedy Gaussian process regression[A]. Advances in Neural Information Processing Systems 13[C]. MIT Press, 2000, 1-7.

[143] Gramacy, R.B., Apley, D.W.. Local Gaussian process approximation for large computer experiments[J]. Journal of Computational and Graphical Statistics, 2015, 24(2): 564-578.

[144] Schreiter, J., Englert, P., Nguyen-Tuong, D., Toussaint, M.. Sparse Gaussian process regression for compliant, real-time robot control[A]. 2015 IEEE International Conference on Robotics and Automation[C]. IEEE, 2015, 2586-2591.

[145] Herbrich, R., Lawrence, N., Seeger, M.. Fast sparse Gaussian process methods: the informative vector machine[A]. Advances in Neural Information Processing

Systems 15[C]. MIT Press, 2002, 1-8.

[146] Seeger M., Williams, C.K.I., Lawrence, N.D.. Fast forward selection to speed up sparse Gaussian process regression[A]. Ninth International Workshop on Artificial Intelligence and Statistics[C]. AISTATS, 2003, 1-8.

[147] Wang, W., Zhou, C.. A two-layer aggregation model with effective consistency for large-scale Gaussian process regression[J]. Engineering Applications of Artificial Intelligence, 2021, 106: 104449.

[148] Jakkala, K.. Deep Gaussian processes: a survey. arXiv preprint arXiv: 2106.12135, 2021.

[149] Havasi, M., Hernandez-Lobato, J.M., Murillo-Fuentes, J.J.. Inference in Deep Gaussian Processes using Stochastic Gradient Hamiltonian Monte Carlo[A]. 32nd Conference on Neural Information Processing Systems[C]. NeurIPS, 2018, 1-11.

[150] Rasmussen, C.E., Nickisch, H.. Gaussian Processes for Machine Learning (GPML) Toolbox[J]. Journal of Machine Learning Research, 2010, 11: 3011–3015.

[151] Hamidieh, K.. A data-driven statistical model for predicting the critical temperature of a superconductor[J]. Computational Materials Science, 2018, 154: 346-354.

[152] Dua, D., Graff, C., 2019. UCI Machine Learning Repository[DB/OL]. http://archive.ics.uci.edu/ml.

[153] Neshat, M., Alexander, B., Sergiienko, N.Y., Wagner, M.. New insights into position optimization of wave energy converters using hybrid local search[J]. Swarm and Evolutionary Computation, 2020, 59: 100744.

[154] Neshat, M., Alexander, B., Wagner, M., Xia, Y.. A detailed comparison of meta-heuristic methods for optimising wave energy converter placements[A]. 2018 Genetic and Evolutionary Computation Conference[C]. GECCO, 2018, 1318–1325.

[155] Bertin-Mahieux, T., Ellis, D.P.W., Whitman, B., Lamere, P.. The million song dataset[J]. 12th International Society for Music Information Retrieval Conference[C]. ISMIR, 2011, 24–28.

[156] Burgués, J., Jiménez-Soto, J.M., Marco, S., 2018. Estimation of the limit of detection in semiconductor gas sensors through linearized calibration models[J]. Analytica Chimica Acta, 2018, 1013: 13–25.

[157] Burgués, J., Marco, S., 2018. Multivariate estimation of the limit of detection

by orthogonal partial least squares in temperature-modulated MOX sensors[J]. Analytica Chimica Acta, 1019: 49–64.

[158] Brooks, C., Burke, S., Persand, G.. Benchmarks and the accuracy of garch model estimation[J]. International Journal of Forecasting, 2001, 17: 45–56.

[159] Charles, A., Darne, O., 2019. The accuracy of asymmetric garch model estimation[J]. International Economics, 2019, 157: 179–202.

[160] Pereira, F.C., Antoniou, C., Fargas, J.A., Ben-Akiva, M.. A metamodel for estimating error bounds in real-time traffic prediction systems[J]. IEEE Transactions on Intelligent Transportation Systems, 2014, 15: 1310–1322.

[161] Urban, S., Ludersdorfer, M., Patrick, V.D.S.. Sensor calibration and hysteresis compensation with heteroscedastic gaussian processes[J]. IEEE Sensors Journal, 2015, 15: 6498–6506.

[162] Munoz-Gonzalez, L., Lazaro-Gredilla, M., Figueiras-Vidal, A.R.. Divisive gaussian processes for nonstationary regression[J]. IEEE Transactions on Neural Networks and Learning Systems, 2014, 25: 1991–2003.

[163] Munoz-Gonzalez, L., Lazaro-Gredilla, M., Figueiras-Vidal, A.R.. Laplace approximation for divisive gaussian processes for nonstationary regression[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38: 618–624.

[164] Saul, A.D., Hensman, J., Vehtari, A., Lawrence, N.D.. Chained gaussian processes[A]. 19th International Conference on Artificial Intelligence and Statistics[C]. PMLR, 2016, 51: 1431-1440.

[165] Goldberg, P.W., Williams, C.K.I., Bishop, C.M.. Regression with input-dependent noise: A gaussian process treatment[A]. Advances in Neural Information Processing Systems 10[C]. NeurIPS, 1997, 493-499.

[166] Kersting, K., Plagemann, C., Pfaff, P., Burgard,W.. Most-likely heteroscedastic gaussian process regression[A]. 24th International Conference on Machine Learning[C]. Association for Computing Machiner, 2007, 393–400.

[167] Heinonen, M., Mannerstrom, H., Rousu, J., Kaski, S., Lhdesmki, H.. Non-stationary gaussian process regression with hamiltonian monte carlo[A]. 19th International Conference on Artificial Intelligence and Statistics[C]. PMLR, 2016, 51: 732-740.

[168] Binois, M., Gramacy, R.B., Ludkovski, M.. Practical heteroscedastic gaussian process modeling for large simulation experiments[J]. Journal of Computational

and Graphical Statistics, 2018, 27: 808– 821.

[169] Zhang, Q.H., Ni, Y.Q.. Improved most likely heteroscedastic gaussian process regression via bayesian residual moment estimator[J]. IEEE Transactions on Signal Processing, 2020, 68: 3450–3460.

[170] Lázaro-Gredilla, M., Titsias, M.K.. Variational heteroscedastic gaussian process regression[A]. 28th International Conference on International Conference on Machine Learning[C]. Omnipress, 2011, 841-848.

[171] Menictas, M., Wand, M.P.. Variational inference for heteroscedastic semiparametric regression[J]. Australian & New Zealand Journal of Statistics, 2015, 57: 119–138.

[172] Munoz-González, L., Lázaro-Gredilla, M., Figueiras-Vidal, A.R.. Heteroscedastic gaussian process regression using expectation propagation[A]. 2011 IEEE International Workshop on Machine Learning for Signal Processing[C]. IEEE, 2011, 1-6.

[173] Tolvanen, V., Jylanki, P., Vehtari, A.. Expectation propagation for nonstationary heteroscedastic gaussian process regression[A]. 2014 IEEE International Workshop on Machine Learning for Signal Processing[C]. IEEE, 2014, 1-6.

[174] Hartmann, M., Vanhatalo, J.. Laplace approximation and natural gradient for gaussian process regression with heteroscedastic student-f model[J]. Statistics & Computing, 2019, 29: 753–773.

[175] Gittens, A., Mahoney, M.. Revisiting the Nystrom method for improved large-scale machine learning[A]. 30th International Conference on Machine Learning[C], PMLR, 28(3): 567-575, 2013.

[176] Alaoui, A.E., Mahoney, M.W.. Fast randomized kernel ridge regression with statistical guarantees[A]. 28th International Conference on Neural Information Processing Systems[C]. MIT Press, 2015, 775–783.

[177] Burt, D.R., Rasmussen, C.E., Mark, V.D.W.. Convergence of sparse variational inference in gaussian processes regression[J]. Journal of Machine Learning Research, 2020, 21: 1–63.

[178] Ji, C., Shen, H.. Stochastic variational inference via upper bound. arXiv preprint arXiv:1912.00650, 2019.

[179] Dieng, A.B., Tran, D., Ranganath, R., Paisley, J., Blei, D.M.. Variational inference via $\mathcal{X}$-upper bound minimization[A]. 31st International Conference on Neural Information Processing Systems[C]. Curran Associates Inc., 2017,

2729-2738.

[180] Brooks, F.T., Pope, D.S., Marcolini, A.M.. Airfoil self-noise and prediction[R]. Technical Report, NASA-RP-1218, 1989.

[181] Kaya, H., Tüfekci, P.. Local and global learning methods for predicting power of a combined gas & steam turbine[A] Intertational Conference on Emerging Trends in Computer and Electronics Engineering[C]. ICETCEE, 2012, 13-18.

[182] Tüfekci, P.. Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods[J]. International Journal of Electrical Power & Energy Systems, 2014, 60: 126-140.

[183] Kaya, H., Tüfekci, P., Uzun, E.. Predicting co and noxemissions from gas turbines: novel data and abenchmark pems[J]. Turkish Journal of Electrical Engineering & Computer Sciences, 2019, 27: 4783–4796.

[184] MacKay, D.J.. Neural Networks and Machine Learning[M]. Springer-Verlag, 1998, 133-165.

[185] Chen, X., Wang, B.. How priors of initial hyperparameters affect Gaussian process regression models[J]. Neurocomputing, 2018, 275: 1702-1710.

[186] Teng, T., Chen, J., Zhang, Y., Low, B.K.H.. Scalable Variational Bayesian Kernel Selection for Sparse Gaussian Process Regression[A]. 34th Conference on Artificial Intelligence[C]. AAAI, 2020, 34(4): 5997-6004

致 谢

本人于杭州电子科技大学度过本科加硕士接近七年的时间，得到了许多帮助，借此机会表达诚挚的感谢：

首先，感谢导师王文胜教授，王老师的学识渊博本身就是良好的学习对象，在初入研究生期间指点了这个极具研究价值的模型，并且讨论期间提供的建议具备良好的合理性与参考价值，也在研究外提供了非常大的帮助。感谢家人们，为本人创造了非常优秀的学习条件。

同时，感谢经济学院的老师们，尤其是斯介生老师的建议与解答，和郑静老师在讨论班上的问题与建议。也感谢师姐师兄师弟师妹们，感谢研究生同班以及同学院的小伙伴们，以及本科时期授过课的理学院老师们。

其次，感谢浙江华为公司提供的实习机会，并且本文的第二章与第三章内容中大部分工作皆于实习工作之余完成，变相得到资助。同时感谢组内李滔与苏宝星在工作上提供的帮助。

然后，感谢本人读研期间在网络上认识的陈平、温铁军、翟东升、艾跃进等老师，其思想深刻，犹如高屋建瓴，当为本人的学习榜样。具体的，本人于 2019 下半年开始接触陈平老师主持的“眉山论剑”栏目，当时关于“修例风波”的见解着实使本人耳目一新，之后便一直关注该老师，也翻阅了老师的论文集及出版物，引起了本人对生命科学与复杂系统的兴趣，颇具拨云见日之效。另一位同样使本人产生“熠熠生辉”感觉的好友曹宁于今年在莫斯科大学进修力学数学博士，在此祝愿他顺利毕业，也感谢他在本科与硕士期间和本人分享、交流观点。

再次，感谢华红光教练从大一开始提供的帮助以及其培养的锻炼习惯，终身受益。感谢刘海涛教授扎实的工作，本人硕士期间的研究内容主要基于其工作。感谢吴安琪教授，虽仅有一信交流，且本人也未满足 GT 的申请条件，但于 21 年 6 月关注到您的信息及研究领域，是本人完成本文第二章与第三章的主要动力。

感谢好友黄家杨、孔祥昊、周尹茜于论文的语言方面修改提供的帮助，促使本人发表了人生中第一篇论文。感谢好友尹卓立、曹宁在申请方面提供的帮助。感谢好友吴雨璇在语言学习方面提供的帮助。感谢好友卢从安在考研时提供的帮助。同样感谢其他在本科与硕士期间认识的小伙伴们。

最后，感谢百忙之中评阅论文和参加答辩的各位专家、教授。也感谢阅读本文的学生、学者们，因本人水平有限，故在文章细节处理方面不是非常到位，但也愿效抛砖引玉之力，希望对您的研究有所帮助。

附录 作者在读期间的主要学术成果

[1] Wang, W., Zhou, C.. A two-layer aggregation model with effective consistency for large-scale Gaussian process regression[J]. Engineering Applications of Artificial Intelligence, 2021, 106: 104449. Impact Factor: 6.212

[2] Wang, W., Zhou, C.. Variational model selection of inducing points in sparse heteroscedastic Gaussian process regression[J]. Knowledge-Based Systems, 2021, Under Review. Impact Factor: 8.038