基于个性化LSTM的矩阵分解在线QoS预测

摘要

服务质量预测是服务计算中的一个重要课题，在过去十年中得到了广泛的研究，提出了许多时间感知的QoS预测方法，取得了令人激动的预测效果。但是，它们没有提供有效的模型更新机制，因此必须定期对整个模型进行重新训练，以处理即将到来的数据。如何及时更新预测模型，准确预测候选服务的QoS缺失值，成为亟待解决的问题，本文提出了一种基于矩阵分解的基于个性化LSTM的在线QoS预测方法。我们的方法可以捕获多个用户和服务的动态潜在表示，并且可以及时更新预测模型来处理新数据，在实际数据集上进行的实验表明，我们的方法在在线预测性能方面优于几种最先进的方法。

1. INTRODUCTION

随着面向服务的计算（SOC）的广泛发展，越来越多的软件应用程序通过重用现有的Web服务来开发，这些Web服务可以被发现和组合，以交付需要的功能，从而加速Web服务或Web API的创建和部署。越来越多的Web服务为开发人员在构建基于服务的系统中选择合适的服务带来了沉重的负担。服务选择的最终目标是找到能够同时满足用户功能和非功能需求的服务。随着功能相似服务的不断增加，服务的非功能属性因此成为决策服务选择的主要考虑因素。

服务质量QoS指的是Web服务的非功能特性，如响应时间、吞吐量等。它作为确定可以选择或推荐哪些服务的基础。一般来说，QoS值可以在提供方端和用户端上测量。由于服务用户的动态网络环境和不同的地理位置，不同用户在调用相同的服务时可能会观察到不同的QoS值，用户端QoS评估应该得到更多的关注，它与web服务上的个性化用户体验更加相关。获取用户端QoS值的直接方法是让用户进行实际调用，这在实践中是不可行的，因为过程耗时过多开销过大。因此，如何准确地预测不同用户的服务的未知QoS值成为服务计算中的一个迫切问题。

近年来，许多QoS预测方法采用根据Web服务的历史调用记录来预测未知的QoS值。一般来说，由于动态网络环境和服务状态，QoS值随时间的变化很大。基于目标用户服务请求时间来预测QoS值的时间感知QoS预测得到了越来越多的关注。针对这一问题已经提出了许多方法，取得了令人鼓舞的预测效果。在真实的QoS预测场景中，随着时间的推移，新的QoS观测不断地出现。是否需要更新经过训练的预测模型，以及如何更新，在现有文献中还没有得到充分的讨论。为此，我们首先进行了实证实验，分析了现有模型更新时间感知方法的性能。实验中使用的数据集是WSDream的一种变体，它是最流行的QoS预测数据集，它由140个用户在64个时间段内观察到4500个服务的QoS数据组成。更多关于数据集、实验设置和评估度量的描述在第五部分中可以找到。在这里，我们只是说明了对两种代表性的时间感知QoS预测方法的一些观察：CLUS 和 WSPred

从表一中，我们可以观察到，使用最新观察结果的更新模型优于仅利用历史观察结果的原始模型。因此，最近的QoS经验将导致更准确的预测性能，这进一步证实了[7]提出的假设。

从表二中，我们可以看出，这两种利用历史数据进行训练的时间感知方法，优于传统的非时间感知方法，它只利用最新的QoS数据，此外，在44-63之间的平均改进百分比（在训练中积累了更多的历史数据）更好。

根据实验可得，预测模型应该通过增加新出现的数据进行增量更新，以获得更好的预测性能。然而现有的时间感知方法很少提供有效的模型更新机制，他们必须使用大量的历史数据对模型进行训练，这使得他们很难满足在线预测的要求。因此，如何及时更新预测模型，提供在线QoS预测成为了一个重要问题。为了解决这一关键挑战，本文提出了一种基于矩阵分解的新型个性化LSTM算法，用于在线QoS预测。我们的模型可以用来描述多个用户和服务的动态潜在表示，并且可以根据当前的观察和有限的历史数据以及一些长期保留的信息进行更新。本文的主要贡献总结如下：

我们研究Web服务的在线QoS预测问题，这在现有文献中还没有充分讨论。

我们提出了一种新型的基于矩阵分解的个性化LSTM算法，用于在线QoS预测。

对真实数据集进行的实验表明，我们的方法在在线预测性能上优于几种最先进的QoS预测模型，论文的其余部分如下所示，第二节讨论相关工作，准备工作在第三部分介绍，第四节介绍了所提出的在线QoS预测方法。实验结果和分析在第五节给出，最后，第六部分对论文进行了总结，并提出了今后的工作。

1. RELATED WORK

作为动态环境下Web服务推荐和选择的关键技术，QoS预测在过去十年中得到了深入的研究。针对这一问题的现有研究可以分为两类，即是否可以处理时间序列的QoS数据。本领域的第一组研究主要集中在静态时间间隔下的个性化QoS预测，不利用过去的QoS预测。协同过滤方法是该场景中应用最为广泛的技术，可以大致分为两类：基于邻居的和基于模型的。基于邻居的方法根据用户过去的QoS体验度量用户或服务的相似性，然后利用相似邻居的QoS信息，对候选Web服务上的用户进行个性化的QoS预测。基于领域的协同过滤方法可以进一步分为基于用户的、基于服务的和混合的方法。基于模型的方法采用机器学习技术来学习基于训练数据的未来预测模型。基质因素化是这种类型的代表，例如CloudPred和NMF。已有文献表明，基于MF的方法在QoS预测性能上优于基于邻居的方法，然而，这些方法不能捕获QoS值的时间动态，这使得它们很难获得理想的结果。

另一组研究，时间感知的QoS预测方法，训练模型以适应过去的QoS值，然后预测它们的未来趋势。例如，Godse等人提出了基于ARIMA模型来预测服务性能。Amin等提出了一种QoS预测方法，将ARIMA与GARCH模型集成在一起，以解决ARIMA恒定变化假设的局限性。然而，基于ARIMA的模型主要关注于预测单个用户观察到的每个Web服务的QoS值，这使得它们难以预测多个用户在服务选择和个性化推荐过程中观察到的多个服务。因此引入协同过滤方法（CF）来缓解这个问题。例如Hu等人提出的模型是将CF与基于ARIMA和Kalman滤波的改进时间序列预测相结合的模型。[6]、[13]、[14]使用用户服务时间张量对时间感知QoS预测问题进行形式化，并进行张量分解。然而，当维数变大时，张量分解就变得难以处理和低效。Silic等人提出将历史QoS数据进行聚类，并对每个簇的平均信度进行哈希用于预测。Zhu等人采用了类似的策略，提出了一种新的具体情境的矩阵分解方法。

正如第一节所分析的，现有的时间感知方法没有提供有效的模型更新机制。他们必须定期重新训练他们的模型来处理新的数据。受到循环推荐网络和协作RNN网络的启发，我们提出了一种新的变体LSTM：个性化LSTM并使用它作为构建PLMF方法的基本快，它可以捕获用户和服务的时间依赖性和在线预测QoS值。

此外，Wang等人将经典的LSTM应用于单个用户观察到的单个服务的在线可靠性预测。与我们的协同预测多个用户的多个服务的QoS相比，他们的工作很难同时预测多个服务的QoS值，因此很难满足服务选择的需要。

1. PRELIMINARIES

在本节中，我们简要介绍了本文将要用到的一些初步知识。

1. 矩阵分解

如第二节所述，基于矩阵分解（MF）的技术在Web服务QoS预测中得到广泛的应用。给定一个用户服务QoS观测值矩阵 R，UxS ，U是用户，S是服务，MF学习用户潜在表征 ，服务潜在表征 ，定义QoS值 为

C:\Users\cxq\AppData\Roaming\Tencent\Users\928894482\QQ\WinTemp\RichOle\KIJ65R1~_WB0Q4@K[H6][@3.png

F表示和的交互功能表示F的参数，通常F可以表示两个向量的内积。

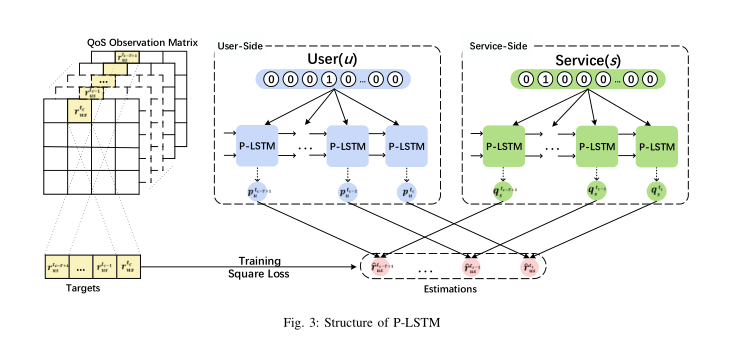
1. 长短时记忆网络

长短时记忆网络是递归神经网络的一种变体，它可以从序列中捕获时间动态，与传统的RNN相比，LSTM引入了一种门禁单元，它由输入门、忘记门、输出门和存储单元状态组成。在每个时间间隔t,如果输入门已经被激活，新的输入连同最新的单位输出h(t-1)将转化和积累到细胞状态中。同样地，最新的细胞状态会被遗忘门控制遗忘信息。输出门决定了什么样的信息将被输出。

在QoS预测场景中，可以像在[16]中那样使用经典的LSTM模型来学习序列波动模式。但是，正如前面所提到的，它不能捕获不同用户或不同服务之间的潜在差异，从而导致性能低下。因此，我们提出了一种新型的个性化的LSTM模型来预测QoS。

1. OUR APPROACH
2. 总体框架
3. 问题陈述：假设有M个用户U={u1, u2, uM} 和 N个Web服务 S={s1,s2, sN}。当用户调用Web服务时，他可以从自己的角度观察服务的QoS值。通过这种方式，每个服务都有不同用户观察到的不同QoS值，这些值可能随时间而变化。如果用户调用服务 ,此服务的QoS值调用将被记录下来。在给定当前时间间隔的用户服务请求的情况下，本文要解决的问题是如何及时、准确地预测候选服务在服务选择和推荐中缺失的QoS值。
4. 基本解决方案：正如第一节所讨论的，现有的方法不能将最新的QoS观察纳入到他们的预测模型中，这阻碍了它们提供理想的在线预测性能。本文旨在提出一种在线QoS预测方法来解决这一问题。图3显示了所提方法的框架。

在训练阶段，我们的模型试图捕获用户和服务之间的时间依赖性。如图1所示，表示当前时间间隔和表示比提前T-1个间隔的时间。在时刻，我们的方法利用了当前观察值，一些历史数据，随着长期保留信息，更新最新的模型。在接下来的时间间隔，成为和整个计算往前移动一个时间间隔，可视为一个滑动窗口长度t, 随着时间的推移，模型将会定期更新。在预测阶段，给定用户的服务请求，可以根据最新的训练模型计算所有候选服务的个性化QoS预测。



1. 利用MF对QoS进行在线预测分析

由于矩阵分解在QoS预测中表现出来明显的优势，我们也尝试在在线QoS预测场景中采用MF的思想。根据MF范式，用户u在时间间隔t时调用的服务s的QoS值可以通过u和s的潜在表示的组合来估计，如式2所示：

C:\Users\cxq\AppData\Roaming\Tencent\Users\928894482\QQ\WinTemp\RichOle\%1%GE]`ZHO`OTB@)0DW4KL9.png

其中 和 分别表示时间间隔为t的用户u和服务s的潜在表示，F为交互函数，为F的参数。 和  分别表示用户和服务的潜在表示矩阵。如果采用经典的MF技术，则 的计算完全依赖于，即t处的QoS观察值，但该策略不能充分利用以往的观测结果，也不能捕获时间相关性，从而导致预测性能较差。由于LSTM在捕获时间依赖性方面的优势，我们使用两个LSTM来分别学习 和，以克服这些局限性。

1. 个性化LSTM

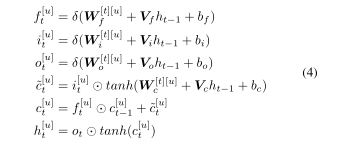
为了描述多用户和服务在QoS预测中的动态潜在表现，我们提出了LSTM的一个新变体：个性化LSTM（P-LSTM）

描述在图4中，P-LSTM和古典LSTM之间的关键区别在于，忘记门、输入门、输出门和内部记忆状态的权重矩阵表示为嵌入矩阵，可视为嵌入用户和服务的查找表。它们的输出可以看做是用户或服务的密集表示，类似于矩阵分解的潜在因素。通过这种方式，这些结构变得个性化，其输出因用户或服务而异。,,,表示用户u对应的个性化结构。以遗忘门为例，的嵌入矩阵定义为：

C:\Users\cxq\AppData\Roaming\Tencent\Users\928894482\QQ\WinTemp\RichOle\LX}XT)6)E$9DM6GH65}I998.png

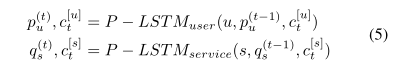
其中 维度为K, K表示嵌入矩阵的维度，M是用户的数量， 表示用户u的遗忘门嵌入，当用户u输入时，其忘记门嵌入将输出和与最新输出参与的计算。从本质上来说，决定遗忘什么信息。

正式地说，用户u的个性化LSTM的计算过程如下：

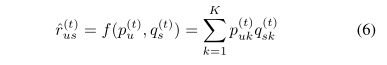


其中，为t时个性化LSTM的输出，为细胞状态，W和v为权重矩阵，b为偏置项，为Hadamard的乘积，对于服务s，计算过程相似。

1. 基于个性化LSTM的矩阵分解
2. 模型训练：PLMF结构如图3所示，请注意，在每个时间间隔中，PLMF使用滑动窗口中T时间间隔的观察以及长期内存来更新最新模型。给定用户u和服务s, 模型将估计C:\Users\cxq\AppData\Roaming\Tencent\Users\928894482\QQ\WinTemp\RichOle\{C}4QROWG}5S{T$_M303UNB.png进行训练。我们首先关注，在t时刻的估计值，然后利用滑动窗口扩展到整个估计过程，然后再进行模型的优化。在最上面的输入层，u和s的标识符首先被转换为具有热编码的稀疏二进制向量，然后将它们分别输入到两个P-LSTM中：用户端P-LSTM和服务端P-LSTM，以学习动态潜在表示。 和 分别表示u和s的潜在表示， 分别表示u和s的内部状态，动态潜在表示可以定义为：



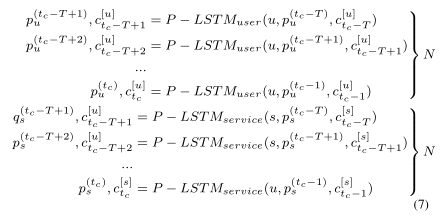
其中P-LSTM user 和P-LSTM service表示用户端和服务端的计算过程。一旦获得用户和服务的潜在表示形式，我们就可以按照矩阵分解范式估计QoS值：

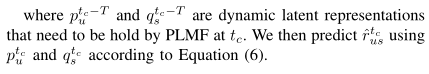


其中，K表示潜在表示向量的维数，请注意，我们还可以采用其他非线性映射，如[18],考虑计算复杂度的要求，我们采用和的线性组合来估计

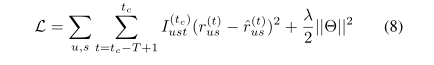
为了捕获更可靠的时间依赖性，可以将历史QoS数据合并到每个时间间隔的模型训练中，如前所述，在时间轴上有一个大小为T的滑动窗口。在每一个新的时间间隔，滑动窗口向前移动一个时间间隔，窗口内的QoS观测值作为训练集。模型根据下式估计

C:\Users\cxq\AppData\Roaming\Tencent\Users\928894482\QQ\WinTemp\RichOle\)EZ2_06O)`Z[UA0}N$GUJ]M.png





在此设置下，优化目标是找到T时间间隔内接近实际QoS观测值的参数。我们采用均方根误差作为损失函数，用L2正则化控制模型的复杂度。将代价函数L最小化的定义如下：



BPTT在现有文献中得到了广泛的应用，用于优化LSTM模型，但是，在我们的问题中不适合直接应用BPTT，因为QoS的预测同时依赖于两个P-LSTM，为了缓解这个问题，我们首先修复用户端参数，并遍历训练集以更新服务端P-LSTM，然后我们修复服务端参数，并遍历训练集以更新用户端P-LSTM。利用这种策略，我们可以选择对用户端P-LSTM和服务端P-LSTM进行优化，直到模型收敛或达到早期停止状态。此外，我们使用自适应矩估计Adam来学习我们提出的模型。

1. 预测：在tc使用PLMF训练模型，我们可以利用式7计算u和s的动态潜在表征，然后根据式6估计QoS值。由于QoS值非负，所以我们调整负预测为零。
2. EXPERIMENTS

在这一节中，我们进行了一系列的实验来评价所提出的方法。这些实验旨在回答以下研究问题：

RQ1：在预测性能方面，该方法是否优于最先进的QoS预测方法？

RQ2：我们模型的更新时间是否比现有的时间感知QoS预测方法更短？

RQ3：增加滑动窗口的大小是否可以提高QoS预测性能？

1. 实验设置：

所有的实验程序都是在GPU工作站上进行的。我们在Pytorch1(一种广泛采用的深度学习框架)的基础上实现了我们的方法，并在GPU上运行。对于相互竞争的方法，我们采用了它们的公共可用版本2，在c++中实现。

1. 数据集：WSDream是一个真实的Web服务QoS数据集，是QoS预测中最流行的数据集。WSDream的时间序列QoS数据有两个限制。首先，一些用户QoS数据过于稀疏，无法进行预测，在第133和第126位用户的稀疏性分别为99.98%和99.97%。其次，数据集缺乏关于用户和服务的侧信息，例如用户位置、ip和ASNs，这不利于后续的研究。考虑到这些弱点，我们基于WSDream的原始观察创建了一个新的数据集，删除具有极稀疏数据的用户，并添加用户和服务的附加信息。新的数据集包括140个用户在64个时间间隔内观察到的4500个服务的响应时间，总共有27392643条记录，稀疏性约为66.98%。本文使用的源代码和数据集可以在我们的项目页面中检索。
2. 评价指标：我们采用平均绝对误差MAE和均方根误差RMSE来度量预测性能。
3. 竞争方法：为了评估我们的方法的有效性，我们选择了几种最先进的QoS预测方法进行比较。在这些方法中，Cloud-Pred 和 NMF是非时间感知的，而WSPred 和CARP都是时间感知的。由于这些方法都不是自然的在线模型，为了与我们的PLMF进行公平的比较，我们允许这些方法利用最新的QoS数据。详细的策略描述如下：假设在时刻t时，历史数据和当前数据记为 R1，……，Rt 。首先，每个区间的QoS数据被分为两组：训练集合测试集，根据预先定义的比率，定义

C:\Users\cxq\AppData\Roaming\Tencent\Users\928894482\QQ\WinTemp\RichOle\W)_X$C$IP$MEW4D}63MOVE4.png

对于时间感知的方法，,……都作为训练集，对于非时间感知的方法，只有当前观察值作为训练集。对于所有的方法， 作为测试集。训练和测试过程在每个时间间隔进行评估。

1. 参数设置：我们随机抽取训练的10%构建交叉验证集并相应调整超参数。模型参数按高斯分布随机初始化。根据参数调优结果，我们使用完整的批处理学习，设置P-LSTM学习速率α，正则化参数 和 dropout 比例分别为0.02,30.0， 0.25。在第五节B部分，我们分别将T设置为3，潜在表示的维度K为80，在第五节的C部分和D部分，我们设置K为64,80,80，96 而T=1,2,3,4.我们还发现，在模型中加入预先训练过的潜在表示可以获得更好的性能。潜在表示由预先训练的NMF初始化。

对于竞争的方法，我们通过遵循WSDream的默认设置来调整它们的超参数。

1. 性能比较（RQ1）

在现实世界中，观察到的QoS数据非常稀疏，因为每个用户只调用少量的服务。在我们的实验中，我们设置了不同的数据密度来构建一个数据集，从5%到20%，步长大小为5%，比较结果见表三。

表三中给出了QoS预测性能的比较结果。注意，每个值都是所有区间的加权平均值。我们可以观察到，我们的PLMF方法取得了优于其他方法的较小的MAE和RMSE，特别是，当密度增大时，PLMF的性能得到了较大的提高，这说明了我们的方法的表达能力。

为了比较每一个区间的详细性能，图4显示了所有方法在训练集上的详细性能，每间隔1-63，密度为20%，如图4所示，所有方法的性能随时间间隔呈现相似的变化趋势，但CARPs的性能变化幅度更大。与非时效性的NMF和CloudPred相比，PLMF在任何时候都优于它们。与时间感知方法相比，PLMF在大多数时间间隔都达到了最佳，特别是数据相对稀疏的情况下。它表明，传统的时间感知方法需要大量的数据才能获得相对更好的性能。请注意，在每个时间间隔中，时间感知方法不仅利用数据同时也对所有的历史数据进行了新的训练。相比之下，PLMF只使用当前时间间隔的数据和几个时间间隔的历史数据（在这组实验中，使用了两个间隔的历史数据）更新最新的模型。例如，在时间间隔40中，时间感知方法使用40个间隔（从1到40）的所有数据来训练一个新模型，而PLMF只使用3个间隔（从38到40）的数据。从第5节C部分可以看出，随着训练数据的增加，在模型上花费的训练时间显著增加，远远超过PLMF。

1. K的影响：如前所述，PLMF通过学习用户和服务的潜在表现形式，采用矩阵分解范式进行QoS预测，其大小为K，因为K控制模型的复杂性，其设置是一个关键问题。如果K太大，模型可能会陷入过度拟合，从而降低其泛化能力。如果K太下，则模型可能不太适合数据，因此无法描述用户和服务的特征。为了研究K对QoS预测的影响，我们将K设为从32到80，步长为8.图5为不同K的MAE和RMSE，结果表明当K增加到一个合适的范围时，模型的表现能力可以得到增强。虽然K从32增加到72，但是MAE和RMSE的值都显示出接近上升的趋势，尽管K=64时发生了一些波动，当K超过这个范围时，表现为下降趋势。
2. 更新时间比较(RQ2)

在在线学习场景中，训练速度是另一个关键问题，为了研究PLMF是否比其他时间感知方法更新更快，我们选择了三个有代表性的时间间隔（即23,43,63）然后研究了PLMF在这三个区间的测试集上的收敛性。然后，我们比较了训练时间，以达到PLMF和其他方法在这三个区间内的最佳性能。实验进行了10次，取平均值作为结果。图6为PLMF在三个区间的收敛。注意，由于空间限制，我们省略了MAE的结果。我们可以观察到PLMF可以在有限的迭代中获得理想的性能。以区间43为例，该模型经过13次迭代得到了最优的RMSE。此外，在10次迭代中，PLMF获得最佳的RMSE：98.3%、99.9%，98.4% ，间隔分别为23,43，和63。训练时间的比较结果如表4所示，其中PLMF-1, PLMF-2,PLMF-3分别表示PLMF的窗口大小T=1,2,3。由于非时间感知方法的预测性能与其他方法相比不具有可比性，所以我们只选择时间感知方法进行比较。首先，随着T的增加，尽管有一些例外，训练时间呈上升趋势，如在第63区间，PLMF-3的训练时间小于PLMF-1，这个例外可以解释为，虽然单个迭代的训练时间随着T的增加而变长，但是收敛速度可能更快。其次，随着时间间隔的增加，由于数据量的增加，时间感知方法的训练时间显著增加，相比之下，PLMF由于其在线性质，在训练期间保持了良好的稳定性，我们可以得出结论，在在线学习场景中，随着时间的推移，PLMF在训练速度上比其他方法有显著的优势。

1. 更长的滑动窗口有帮助吗（RQ3）

使用滑动窗口，我们的方法可以在每次更新中合并更多的数据，但也可能导致更长的训练时间。理论上，一个较长的滑动窗口可以帮助模型捕获更稳定时间依赖关系。在这个实验中，我们想通过实验来研究一个较长的滑动窗口是否会带来更好的性能。我们将滑动窗口T的尺寸从1调整到4，同时固定其他参数，实验仍在矩阵密度为10%的数据集上进行。从表5中可以看出，当T从1到4时，MAE和RMSE均呈现下降趋势。结果表明，通过增大滑动窗口的尺寸，可以进一步提高预测性能，这是令人鼓舞的。

1. CONCLUSION

本文提出了一种基于矩阵分解的基于个性化LSTM的在线QoS预测方法，我们的方法可以捕获多个用户和服务的动态潜在表示，并且预测模型可以及时更新以处理新出现的数据。在未来，我们计划将我们的方法应用到包括更多QoS属性在内的其他真实数据集，我们还考虑将用户和服务的位置和IP地址等更多的方面信息集成到我们的模型中。