一种基于支持向量回归的互联网端到端延迟预测算法*

钱 峰, 连 涛, 吴嘉兴

(电子科技大学 宽带光纤传输与通信网技术教育部重点实验室, 成都 610054)

摘 要: 互联网端到端延迟是指 IP 分组沿着互联网中一条确定路径进行传输的延迟,端到端延迟的精确预测是大量网络活动的基础,从网络协议设计到网络监测,再从确保端到端 QoS 性能到各种实时业务性能提升。提出一种新的端到端延迟的预测方法,主要贡献有: a)将互联网端到端延迟预测的问题转换为多元回归的预测问题,提出了基于多元回归的端到端延迟预测框架;b)采用支持向量回归 SVR 方法来求解端到端延迟的多元回归问题,提出了基于 SVR 的互联网端到端延迟预测算法。最后使用互联网采集的 RTT 数据来验证提出的算法,实验结果表明,提出的预测算法具有快速和精确特点,是一种适合实际应用的预测算法。

关键词: 互联网; 端到端延迟; 支持向量回归; 预测

中图分类号: TP393 **文献标志码:** A **文章编号:** 1001-3695(2012)05-1850-04

doi:10.3969/j.issn.1001-3695.2012.05.066

Internet end-to-end delay prediction using support vector regression

QIAN Feng, LIAN Tao, WU Jia-xing

(*Key Laboratory of Broadband Optical Fiber Transmission & Communication Networks for Ministry of Education, University of Electronic Science & Technology of China, Chengdu* 610054, *China*)

Abstract: End-to-end packet delay of the Internet is the IP packet transmission delay along a determined path. An accurate end-to-end delay prediction is fundamental to numerous network activities, from protocol design to network monitoring, and from ensure end-to-end QoS to performance enhancement for realtime network applications. This paper presented a novel methodology for predicting end-to-end delay. The major contributions are: a) It converted the end-to-end delay prediction problem into the multivariate regression, and proposed a multivariate regression-based forecasting framework for end-to-end delay; b) It employed support vector regression (SVR) to solve the multivariate regression problem of end-to-end delay, and induced a SVR-based end-to-end delay predicting algorithm. Finally, it used the actual RTT data collected from Internet to validate the proposed algorithm. Simulation results show that the proposed algorithm has fast and accurate prediction characteristics, which is very suit for practical applications.

Key words: Internet; end-to-end delay; support vector regression(SVR); prediction

0 引言

随着实时语音视频等多媒体实时业务的不断出现,网络端到端延迟正成为越来越重要的网络状态参数之一,而网络端到端延迟精确预测已成为大量网络活动的基础。端到端延迟预测有助于网络协议设计、网络监测和网络层析成像;可以帮助动态确定 IP 分组的长度、发送速率、最优路径(和最小延迟相比)、确保端到端 QoS 性能[1];可以广泛用于许多实时网络应用,如多媒体的自适应播放缓冲[2]、提升 VoIP 应用性能[3]、同步和降低语音电话延迟等。

互联网端到端延迟精确预测并不是一件容易的工作,互联网结构的复杂性和异质性导致了网络端到端延迟的非线性和非平稳性(时变)特性[4],使得很难使用一些简单模型或者方法就能够很好解决这样的预测问题。目前已经有许多文献在研究解决端到端延迟精确预测问题。归纳起来,主要方法有两类。a)基于 ARMA 模型的方法。文献[4,5]采用 auto-regressive moving average(ARMA)方法对历史的延迟数据建模,进而来预测新的端到端延迟。文献[6]在文献[4,5]的基础上将 ARMA 模型扩展成为多模型(interacting multiple model,IMM)的 ARMA 框架,从而相比 ARMA 模型进一步提高了预测的精度。这种方法存在问题是 ARMA 模型很难对非平稳性很强的端到端延迟进行精确建模,最重要的 ARMA 模型和多模型的 ARMA 模型只能用于在线的预测。b)采用线性自适应滤波器方法,如 least mean square(LMS)和 recursive least square(RLS)方法[7]。线性自适应滤波器方法对平稳信号的处理效果非常好,但是对端到端延迟这样的非平稳性时间序列的处理效果并不好。

本文从新的角度出发,提出一种网络端到端延迟预测方法。该预测方法的核心是将端到端延迟的预测问题转换为一个多元回归的预测问题,然后选择合理求解方法来求解这样的多元回归预测问题。本文主要通过引入滑动时窗机制将端到

收稿日期: 2011-09-16; **修回日期:** 2011-10-19 **基金项目:** 国家自然科学基金资助项目(60872033);国防预研基金资助项目

作者简介: 钱峰(1978-),男,浙江绍兴人,讲师,博士,主要研究方向为 IP 网络和信号处理(fengqian@uestc.edc.cn);连涛(1987-),男,重庆人,硕士研究生,主要研究方向为网络层析成像;吴嘉兴(1987-),男,硕士研究生,主要研究方向为通信网络和信号处理.

端延迟预测问题转换为多元回归的预测问题,然后再选择使用 SVR[8,9] 算法来求解。选择 SVR 主要是基于以下考虑:a) SVR 使用非线性核函数代替确定性模型方法,避免了将端到端延迟假定为确定性模型;b) 通过控制训练误差和函数平滑性(smoothness),SVR 能够充分利用端到端延迟值的时间相关性和空间相关性(滑动时窗里值与预测值之间关系),从而提高端到端延迟预测的精确性;c) SVR 具有很高的计算效率,很适合在线训练。

1　问题描述和模型

1.1　端到端延迟问题描述

问题模型如图 1 所示。

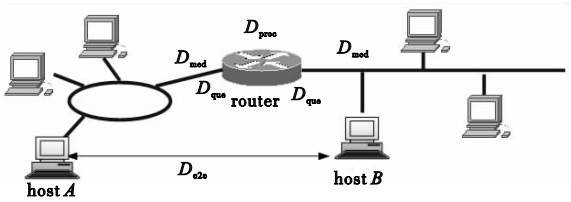

图1　端节点*A*到端节点*B*之间的延迟

如图 1 所示,端到端延迟是两个端节点沿着一条确定的路径所经历的延迟。由于定义的端到端延迟是互联网内的延迟,而不是物理网络内的延迟,因此端到端节点之间需要经过很多中间的路由器。Round trip time(RTT)经常用于研究互联网端到端延迟的变化[10,11],RTT 测量工作只需要在端点主机进行。Jacobson 利用 auto-regressive moving average(ARMA)模型来对 RTT 序列建模和预测,从而设计出了一种拥塞避免算法。利用相同的思路,本文构建一个实验以获得互联网的端到端延迟数据(RTT 值)。在实验中,使用电子科技大学校园内的一台主机作为源节点,然后选择 Google 和 Baidu 等五个不同服务器作为目的主机,通过 ping 程序来收集源节点到不同目的节点的 RTT 值。实验使用了两种不同大小的探测包值,分别为 512 Byte 和 1024 Byte。每个源目的对之间的路径通过周期性使用 tracert 命令来保证路由是稳定的。图 2 为从源节点到目的节点(www.google.com)的 RTT 序列。

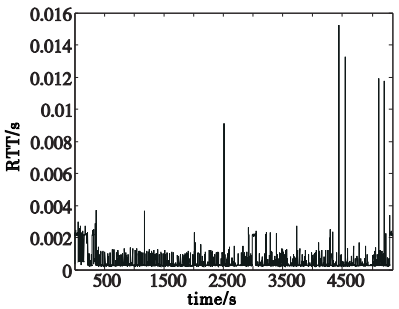

图2　互联网端到端延迟数据(RTT序列)

由于互联网结构的复杂性和异质特性,导致了 RTT 延迟的时变特性。实际上,IP 分组传输过程的 RTT 延迟动态特性受到互联网系统的大量因素的影响,如拥塞、带宽、QoS、传输距离、流量负载和路由协议等。还有部分原因是因为端到端延迟的选择路由需要跨越多个自治系统,而这些自治系统分属于不同管理机构,加剧了端到端延迟的预测复杂性。归纳各种因素,端到端延迟表达式如式(1)所示。

$$D_{e2e}(t) = D_{proc}(t) + D_{que}(t) + D_{med} \tag{1}$$

其中,D_{e2e} 为端到端延迟,如图 1 所示,主要包含三部分内容:路由器处理延迟 D_{proc};路由器队列等待时间 D_{que};物理网络介质的传输延迟 D_{med}。

D_{med} 为物理网络介质的传输延迟,表示 IP 分组在传输介质上进行传输所需要的时间。它主要受介质长度和介质中传输速度两个因素影响。D_{med} 值可以近似认为是常数,而且它的值在整个 D_{e2e} 中所占比率很小。D_{proc} 为路由器的处理延迟,主要受路由器的处理和转发能力、路由协议和 IP 分组头的大小决定。D_{que} 为队列延迟,主要受路由器流量负载影响,IP 网络流量负载具有突发性,所以其为变量。D_{que} 为 D_{e2e} 的主要组成部分,反映了端到端延迟的动态特性。

1.2　端到端延迟的多元回归模型

在收集到 RTT 延迟序列之后,就可以验证算法端到端延迟预测的能力。首先对互联网端到端延迟预测问题建模。如图 3 所示,假设 $\lambda(t)$ 为时刻 t 的 RTT 测量值,设 d 为 RTT 测量值的数目。那么则有 d 维的输入数据 x,其定义为

$$\begin{cases} x = (\lambda(t-(d-1)), \cdots, \lambda(t-1), \lambda(t)) \\ y = \lambda(t+1) \end{cases} \tag{2}$$

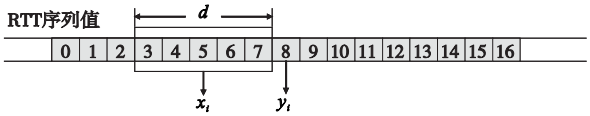

图3　滑动时窗示意图

为了构建所有的可能输入数据 x 值,引入滑动时窗概念,滑动时窗的长度为 d。那么可以在整个时间序列范围构建得到所有可能的输入和输出序列对,可以获得这样的对集合 $\{(x_i, y_i)\}_{i=1,\cdots,L-d}$,这里 L 为时间序列的长度。那么,端到端延迟预测问题就可以转换为一个多元回归问题,其定义为

$$f(x,\omega): x(\mathcal{R}^d) \rightarrow \mathcal{R} \tag{3}$$

其中:ω 为抽象参数集。本文目标是估计得到这样的映射函数 $f(x,\omega)$,从而获得预测值 $\hat{y} = f(x,\omega)$。

多元回归问题求解的方法很多,支持向量回归(SVR)是最典型的方法。为了既能够求解式(2)的问题,又能满足在线预测的需要,本文采用 SVR 算法。SVR 是在多元回归框架内的一种前沿方法,在许多领域有很多成功应用[12]。

2　基于 SVR 的端到端延迟预测算法

2.1　基于 SVR 端到端延迟预测算法

基于 SVR 的端到端延迟预测模型如图 4 所示。

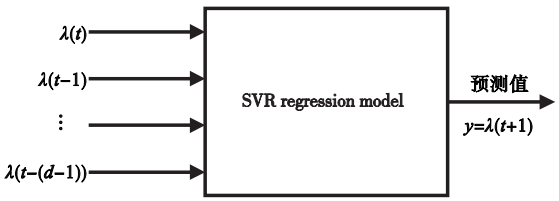

图4　基于SVR的端到端延迟预测模型

基于 SVR 的端到端延迟预测核心就是估计得到式(3)的回归函数 $f_p(\cdot)$。现在已有部分可以用于训练的样本数据 $\{(x_i, y_i)\} \subset \mathcal{R}^d \times \mathcal{R}$,SVR 主要通过样本数据的训练获得回归函数 $f_p(\cdot)$。根据支持向量回归理论,可将此训练过程问题归

结为约束最优化问题[8,9]，其目标函数为

$$\begin{cases} \min \frac{1}{2} \| w \|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) \\ \text{约束信息} \begin{cases} t_i - f(x_i, w) \leqslant \varepsilon + \xi_i^* \\ f(x_i, w) - t_i \leqslant \varepsilon + \xi_i \\ \xi_i, \xi_i^* \geqslant 0, i = 1, \cdots, T \end{cases} \end{cases} \tag{4}$$

其中：C 为用户定义的正则化常数，w 是 l 维权重向量，ε 为一个小的正数，ξ_i 和 ξ_i^* 为非负的松弛变量。式(4)的约束最优化问题可通过线性规划方法求解，得到的解可用式(5)表示。

$$f_p(X) = \sum_{i=1}^{Sv}(\alpha_i^* - \alpha_i)K(X_i, X) + b \tag{5}$$

其中：b 为常数项，α_i 和 α_i^* 为最优化的拉格朗日乘法因子。α_i 和 α_i^* 需要满足以下约束条件：

$$\sum_{i=1}^{Sv}(\alpha_i^* - \alpha_i) = 0, \begin{cases} 0 \leqslant \alpha_i^* \leqslant C \\ 0 \leqslant \alpha_i \leqslant C \end{cases} \tag{6}$$

$K(X_i, X)$ 为核函数，本文使用径向基函数，其定义具体为

$$K(X, X') = \exp(-\gamma \| X - X' \|^2) \tag{7}$$

选择径向基函数的原因主要基于两点：a)径向基函数在时间序列预测中具有非常好的性能；b)在具体实验中，与其他类型的函数进行对比得到。

综上所述，可以得到基于 SVR 端到端延迟预测算法(SVR based end-to-end delay prediction algorithm)，其具体算法可用以下流程来表示。

算法 1

a)提取得到滑动时窗内的已测端到端延迟值序列 x_i 作为算法的输入，同时得到部分历史端到端延迟 y_i；

b)利用线性规划算法求解式(4)的最优化约束问题，得到端到端延迟预测的 SVR 模型 $f_p(\cdot)$；

c)输入新的端到端延迟测量值序列 x_{i+1}；

d)利用式(5)计算得到新的端到端延迟序列值 $\hat{y}$；

e)重复 a)b)，预测所有的端到端延迟值。

2.2 性能量度

使用相对误差(relative error，RE)来估计端到端延迟预测的性能，相对误差的定义为

$$RE = \frac{\sqrt{\sum_{i=1}^{L}(y_i - \hat{y}_i)^2}}{\sqrt{\sum_{i=1}^{L}(y_i)^2}} \tag{8}$$

其中：y_i 为第 i 个时刻真实值，$\hat{y}_i$ 为第 i 个时刻的预测值，L 为时间序列的长度。式(8)为一系列点的相对误差，也可以得到单点的相对误差：

$$RE = \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{9}$$

单点的相对误差主要用来在后面仿真中计算累积量分布函数(cumulative distribution function，CDF)。

2.3 算法性能优化

SVR 算法性能涉及到 SVR 参数和多元回归框架中滑动时窗的优化两方面的优化。其具体优化思路如下：

a)SVR 参数优化。SVR 使用比较困难，因为涉及到非常多的 SVR 性能参数的优化选择。SVR 性能参数的影响主要体现在算法 1 的步骤 b)中。这里主要需要优化的参数是 C 和 γ。本文中，参数 C 和 γ 的优化主要通过交叉测试(cross validation)的方式来进行。

算法 2

a)给定参数 C 和 γ 的范围，然后调用遗传算法进行随机搜索；

b)调用算法 1，计算得到 SVR 拟合误差；

c)重复 a)b)，得到局部最优化 C 和 γ 值。

b)滑动时窗长度 d 的优化。滑动时窗大小也就是输入数据的维数，如果滑动时窗过小，会导致特征量不足，使导致端到端延迟估计精度下降；如果滑动时窗过大，会导致无效的特征量增加，且计算复杂度也增加，所以需要寻找最优的滑动时窗大小。最优滑动时窗大小可以定义为

$$d_{\text{optimal}} = \underset{i \in D}{\operatorname{argmin}}(RE_i) \tag{10}$$

式(10)的求解可以通过下面的算法 3 来完成。

算法 3

a)给定 d 的范围 D，选择不同的 d 值；

b)调用算法 2，预测得到端到端延迟值 $\hat{y}$；

c)利用式(8)计算评价策略 RE 误差；

d)重复 a) ~ c)，直到遍历完选定的 d 的范围，得到最小 RE 误差对应的 d 值，即为最优化的 d 值。

3 实验结果

实验中采用如 2.1 节所描述的直接收集到的 RTT 实际数据值、采集得到 RTT 数据为 6000 个时间序列数据点，其中 3000 个点用于训练，3000 个点用于预测。SVR 算法部分使用 LIBSVM 软件[13]。实验主要分成以下几个部分：a)影响因子的实验评估，主要评估 d 值对预测性能的影响；b)SVR 方法对端到端延迟的非平稳性的追踪能力；c)SVR 算法与 IMM、RLS、LMS 三种方法的性能比较。

a)滑动时窗大小 d 的性能优化。在 $d \in [3, 50]$ 的范围内，对每个点执行 50 次算法 3，以消灭算法的随机性，然后求得平均的 RE 误差。最后得到如图 5 所示的滑动时窗大小 d 的性能优化图。从图 5 可看到，RE 误差随着 d 的增加而急剧减小，然后降到低点，再缓慢回升；滑动时窗大小 d 为 6 时是最佳滑动时窗长度。所以通过优化选择滑动时窗长度 d 可以有效提高端到端延迟预测的性能，同时也说明了算法 3 的有效性。

b)SVR 方法对端到端延迟的非平稳性的追踪能力。根据实验 a)得到的结论(滑动时窗 d 取 6)，调用算法 2 和算法 1 来验证 SVR 算法对端到端延迟的非平稳性的追踪能力。算法 2 得到优化后 $C = 0.073338$ 和 $\gamma = 99.9877$，最后得到如图 6 所示的 SVR 回归方法对端到端延迟的非平稳性的追踪图。从图 6 可以看到，SVR 预测的结果与实际结果总体趋势完全吻合，能够很好地追踪端到端延迟的这种时变特性。所以，SVR 能够比较好地预测端到端延迟。

c)SVR 算法与 IMM-ARMA、RLS 和 LMS 三种方法的性能比较。SVR 算法的设置主要是基于实验 b)和实验 c)的结果。SVR 与 IMM-ARMA 模型、RLS 方法和 LMS 方法的参数设置参照文献[5]。实验主要对 SVR 和 IMM-ARMA 算法进行对比，这是因为 IMM-ARMA 模型比 RLS 和 LMS 方法具有更好预测性能。

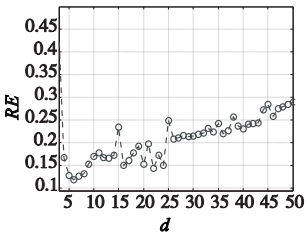

图5　滑动时窗大小*d*的性能优化

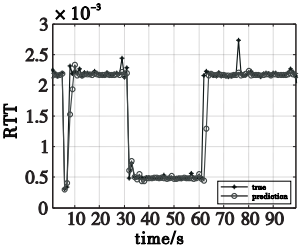

图6　SVR方法对端到端延迟的非平稳性的追踪图

图 7 为 SVR 与 IMM-ARMA 预测性能的对比,中间对角线的斜线为真实值的线。从图 7 可以看到,大部分 SVR 比 IMM-ARMA 更趋近于真实值,只有少量的 SVR 偏移大一些。图 8 为 SVR 与 IMM-ARMA 的 CDF 预测性能对比,这里 CDF 为累计量分布函数的缩写。从图中可以看到 SVR 有 65% 的点是小于等于相对误差 0.2,而 IMM-ARMA 则是 45%。综合图 7 和 8 可以看到,SVR 比 IMM-ARMA 具有更好的预测性能。

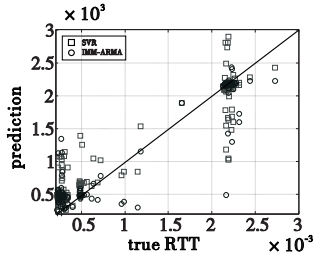

图7　SVR与IMM-ARMA预测性能对比

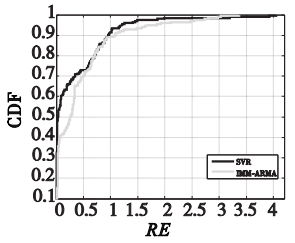

图8　SVR与IMM-ARMA的CDF预测性能对比

将 SVR 方法与 IMM-ARMA 模型、RLS 方法和 LMS 方法进行综合对比,可以得到如表 1 所示的结果。从表 1 可以看到,SVR 的预测性能要好于其他三种方法的预测性能。

表 1　几种预测算法性能比较

	SVR	IMM-ARMA	RLS	LMS
RE/%	18.36	20.21	28.71	35.64

4　结束语

本文提出了一种基于 SVR 的端到端延迟预测算法,并对影响算法性能的因素进行了讨论且提出了优化的方案,从而有效提高了预测算法的性能。最后通过实际采集的数据进行实验,实验分为滑动时窗大小 d 的性能优化、SVR 方法对端到端延迟的非平稳性的追踪能力和 SVR 算法与 IMM-ARMA、RLS、LMS 方法的性能比较三部分,从不同角度验证了本文算法的性能。

后续研究将在上述研究基础上深入和完善,未来倾向于将基于 SVR 端到端延迟预测算法应用于完全在线的预测,这里主要利用在线的 SVR 训练算法,同时研究在线的预测机制,使得算法能够适应完全在线预测的情况。

参考文献:

[1] RAO N S V. Overlay networks of in-situ instruments for probabilistic guarantees on message delays in wide-area networks[J]. IEEE Journal on Selected Areas in Communication,2004,22(1):79-90.

[2] SREENAN C J, CHEN J C, AGMWAL P, *et al*. Delay deduction techniques for payout buffering[J]. IEEE Trans on Multimedia, 2000,2(2):88-100.

[3] KLEPEC K B, TOMAZIC S. Techniques for performance improvement of VoIP applications[C]//Proc of the 11th IEEE Mediterranean MELECON. 2002:250-254.

[4] JACOBSON V. Congestion avoidance and control[C]//Proc of ACM Sigcomm. 1988:314-329.

[5] YANG Ming, LI X R. Predicting end-to-end delay of the Internet using time series analysis [R]. Lakefront:University of New Orleans, 2003.

[6] YANG M, RU J, LI X R, *et al*. Predicting Internet end-to-end delay: a multiple-model approach[C]//Proc of INFOCOM. 2005:2815-2819.

[7] YANG M, RU J F, CHEN H, *et al*. Predicting Internet end-to-end delay a statistical study[J]. Annual Review of Communication, 2005,58(2):665-678.

[8] SCHOLKOPF B, SMOLA A J. Learning with kernels [M]. Cambridge, MA:MIT Press,2002.

[9] SMOLA A J, SCHOLKOPF B. A tutorial on support vector regression [J]. Statistics and Computing,2004,14(3):199-222.

[10] OHSAKI H, MURATA M, MIYAHMA H. Modeling end-to-end packet delay dynamics of the Internet using system identification [C]//Proc of the 17th International Teletraffic Congress. 2001:1027-1038.

[11] PARLOS A G. Identification of the Internet end-to-end delay dynamics using multi-step neuro-predictors[C]//Proc of International Joint Conference on Neural Networks. 2002.

[12] PRIYADARSHINI D, ACHARYA M, MISHRA A P. Link load prediction using support vector regression and optimization [J]. International Journal of Computer Applications,2011,24(7):22-25.

[13] LIBSVM[CP/OL]. http://www.csie.ntu.edu.tw/cjlin/libsvm/.