

## 混合 SDN 的自适应流量估计方法

李晓方<sup>1,2</sup>, 王子磊<sup>1</sup>, 奚宏生<sup>1</sup>

(1. 中国科学技术大学未来网络实验室, 合肥 230027; 2. 中国酒泉卫星发射中心, 甘肃 酒泉 732750)

**摘 要:** 流量矩阵是混合软件定义网络(SDN)流量工程的重要输入,但难以全部直接测量,已有的估计方法主要针对传统 IP 网络,不完全适用于混合 SDN 网络。针对该问题,提出一种基于源-目的(OD)流聚类的自适应多 Elman 神经网络算法。通过对 OD 流按照时间变化模式进行聚类,将单一的高维训练样本分解为多个低维训练样本,强化各低维样本的关键特征,以训练相应的 Elman 神经网络,构成多 Elman 神经网络模型,并利用混合 SDN 中部分 OD 流可以持续精确测量的特点,根据网络状态变化动态调整估计算法的参数。实验结果表明,与广义层析重力算法相比,该算法具有更高的估计精度和更好的自适应能力。

**关键词:** 软件定义网络;流量矩阵估计;神经网络;聚类;K 均值;自适应性

**中文引用格式:** 李晓方,王子磊,奚宏生. 混合 SDN 的自适应流量估计方法[J]. 计算机工程,2016,42(3):103-110.

**英文引用格式:** Li Xiaofang, Wang Zilei, Xi Hongsheng. Adaptive Traffic Estimation Method for Hybrid SDN[J]. Computer Engineering, 2016, 42(3): 103-110.

## Adaptive Traffic Estimation Method for Hybrid SDN

LI Xiaofang<sup>1,2</sup>, WANG Zilei<sup>1</sup>, XI Hongsheng<sup>1</sup>

(1. Laboratory of Future Network, University of Science and Technology of China, Hefei 230027, China;

2. Jiuquan Satellite Launch Center, Jiuquan, Gansu 732750, China)

**【Abstract】** Traffic matrix is an important input for traffic engineering in hybrid Software Defined Network(SDN), but it is difficult to be directly measured. Existing estimation methods are mainly for the traditional IP networks, not fully applicable to hybrid SDN. In order to solve this problem, this paper proposes an Adaptive multi-Elman Neural Network Algorithm (AMElman) based on Origin-Destination(OD) flow clustering. By clustering OD flows according to the time change pattern, the single high-dimensional training sample can be decomposed into several multiple low-dimensional training samples, and each of the low-dimensional samples' key characteristics can be strengthened to train the corresponding Elman neural network and construct the multi-Elman model. At the same time, using the feature that part of the hybrid SDN OD flows can be measured accurately and constantly, the estimation algorithm's parameter can be dynamically adjusted based on change of network status in the estimation process. Experimental results show that this algorithm has higher estimation accuracy and better adaptability than General Tomography(GT) algorithm.

**【Key words】** Software Defined Network(SDN); Traffic Matrix(TM) estimation; neural network; clustering; K-means; adaptivity

**DOI:** 10.3969/j.issn.1000-3428.2016.03.019

### 1 概述

软件定义网络(Software Defined Network, SDN)是近几年学术界和产业界最为关注的网络技术,其核心思想是将控制平面与转发平面分离,以简化网络管理、应用部署等操作。受投资保护、风险控制等因素影响,SDN 的全面部署无法在短期内完成,这将形成

传统 IP 网络与 SDN 共存(Coexist)的状态,即混合 SDN(Hybrid SDN)<sup>[1]</sup>。

流量矩阵(Traffic Matrix, TM)是描述整个网络源-目的(Origin-Destination, OD)间流量大小的工具。在 SDN 中,能够以忽略不计的开销精确测量流量矩阵,以此为输入的流量工程能够极大地提高网络链路的利用率<sup>[2]</sup>。文献[2]将 IP 网络完全改造成

**基金项目:** 国家自然科学基金资助项目(61233003, 61203256); 国家“863”计划基金资助项目(2014AA06A503)。

**作者简介:** 李晓方(1989-), 男, 硕士研究生, 主研方向为软件定义网络、信息安全; 王子磊, 副教授; 奚宏生, 教授。

**收稿日期:** 2015-03-19      **修回日期:** 2015-04-17      **E-mail:** lxf000@mail.edu.cn

SDN 后,网络的链路利用率从原来的约 30% 提高到了近 100%。鉴于 SDN 的这一优势,混合 SDN 的流量工程成为当前的研究热点。混合 SDN 为传统 IP 网络引入了部分集中控制能力,利用 SDN 控制器,能够对部分 OD 流量的路径进行调度和优化,从而提高整个网络的链路利用率<sup>[3]</sup>。但是,混合 SDN 面临一个根本性问题:难以直接获得精确的流量矩阵以作为流量工程的输入,主要因为混合 SDN 中存在着大量 IP 网络节点,相关的 OD 流需要采用估计方法获得。

目前,已有的流量矩阵估计方法主要分为 3 类:(1)统计反演法。文献[4]用 Poisson 分布对 OD 流进行建模,采用网络层析成像法(Network Tomography, NT)来求解。文献[5]用 Gaussian 分布对 OD 流进行建模,采用最大期望法(expectation-maximization, EM)来求解。这类方法由于 OD 流模型过于简单,不能准确描述网络流特征,且先验信息对估计结果的准确性影响很大。(2)附加链路测量信息反演法。文献[6]建立层析成像重力模型对 OD 流进行估计,并用链路的 SNMP 数据进行矫正。文献[7]提出的路由改变方法,通过多次的路由改变,降低问题的病态性,并通过收集的链路 SNMP 数据对噪声进行修正。这些方法通过对链路信息的测量,降低了对先验信息的依赖,但由于缺乏直接测量的 OD 流信息,在整个估计周期内仍存在较大误差。(3)测量反演结合法。文献[8]提出的 Fanout 方法,及文献[9]提出的 PCA 方法,利用实际的链路和 OD 流测量数据,挖掘了 OD 流的时间、空间和时空相关性,进一步消除了估计误差。但这些方法在提高估计准确度的同时,引入了昂贵的测量开销。

上述方法都是面向传统 IP 网络的,尽管混合 SDN 中所要解决的问题与之具有相似性,但这些方法并不完全适用。混合 SDN 中的 SDN 网络设备能够在直接测量代价忽略不计的情况下,精确地持续测量部分 OD 流。本文针对混合 SDN 的这一特点,提出一种基于 OD 流聚类的自适应多 Elman 网络估计算法(AMElman)。

## 2 问题描述

流量矩阵由网络中所有的 OD 流组成,可用一个  $I$  维随机向量  $\mathbf{X} = (X_1, X_2, \dots, X_I)^T$  表示,  $X_i$  表示第  $i$  个 OD 流的流量。令  $J$  维随机向量  $\mathbf{Y} = (Y_1, Y_2, \dots, Y_J)^T$ ,  $Y_j$  表示网络中第  $j$  条链路上的流量;  $\mathbf{A}_{J \times I}$  是网络的路由矩阵,若第  $i$  个 OD 流经过第  $j$  条链路,则  $A_{j,i} = 1$ , 否

则,  $A_{j,i} = 0$ 。在时刻  $t$ , 流量矩阵  $\mathbf{X}$ 、链路流量  $\mathbf{Y}$  和路由矩阵  $\mathbf{A}$  之间的关系表示为:

$$\mathbf{Y}(t) = \mathbf{A}\mathbf{X}(t) \quad (1)$$

混合 SDN 中存在 2 种 OD 流:SDN-OD 流和 IP-OD 流。SDN-OD 流定义为由 SDN 网络设备直接测量的 OD 流,测量开销可忽略不计;IP-OD 流定义为通过传统 IP 网中的流量监控器测量的 OD 流,测量开销巨大。可以认为  $\mathbf{X}$  中的 SDN-OD 流已知,但由于 SDN-OD 流的数量远小于 IP-OD 流<sup>[3]</sup>, 仍存在  $I \gg J$  情况,即  $\mathbf{A}$  是一个非满秩矩阵。在已知  $\mathbf{Y}$  和  $\mathbf{A}$  的条件下,求解  $\mathbf{X}$  是一个病态的线性逆问题。

由式(1)可知,  $\mathbf{Y}$  与  $\mathbf{X}$  之间存在映射关系  $F: \mathbf{X}(t) = F(\mathbf{Y}(t))$ 。要求解  $\mathbf{X}$ , 即需对映射  $F(\cdot)$  进行建模。由于式(1)的病态性,建立关于  $F(\cdot)$  的解析模型是极其困难的。由神经网络理论中的 Kolmogorov 连续性定理<sup>[10]</sup>可知:给定任一连续函数  $\Phi: E^m \rightarrow R^n$ ,  $\mathbf{X} = \Phi(\mathbf{Y})$ ,  $E \in [0, 1]$ , 则  $\Phi$  可以精确地由一个 3 层网络实现。即 3 层神经网络通过学习能够逼近任意的非线性映射,故可利用神经网络对映射  $F(\cdot)$  进行建模。

但是,由图 1 中所示 Abilene 网络中部分 OD 流在一周内的变化情况可见,OD 流具有时变非平稳性,即  $\{\mathbf{X}(t)\}_{t=1}^S$  是一个多变量非平稳时间序列。因此,  $F(\cdot)$  需要具备时变适应能力,才能建立更准确的映射关系。这就要求用于建模的神经网络必须具有时间维度的记忆性。

由于时变非平稳序列  $\{\mathbf{X}_i(t)\}_{t=1}^S$  的变化特征通常受趋势变动、周期变动和不规则变动等情况影响,导致  $\{\mathbf{X}(t)\}_{t=1}^S$  中往往包含大量不同变化模式的时间序列,如表 1 所示。若神经网络在训练过程中同时学习过多的时间序列模式,会造成对各个模式关键规律的敏感性降低。而当前的网络通常具有高维度的  $\mathbf{X}$  和  $\mathbf{Y}$ , 导致神经网络的训练过程非常缓慢。同时,  $\{\mathbf{X}(t)\}_{t=1}^S$  的时变非平稳性,对神经网络模型参数的动态自适应能力提出挑战。

文献[11]提出的多 BP 方法,对若干时刻的链路流量在空间维度上进行分类,以训练每类链路流量所对应的 BP 网络。该方法在一定程度上克服了单 BP 网络学习过多规律时的缺陷,但未能覆盖所有的链路流量状态,也未充分利用网络流量在时间维度上的相关信息,得到的多 BP 网络无法并行运行。文献[12]提出的 TMRI (Traffic Matrix Recurrence Inference) 方法,利用 RBF 神经网络对大尺度 IP 骨干网的流量矩阵进行估计。探讨算法对网络状态变化的适应能力,但未提出自动调整估计模型参数的方法。

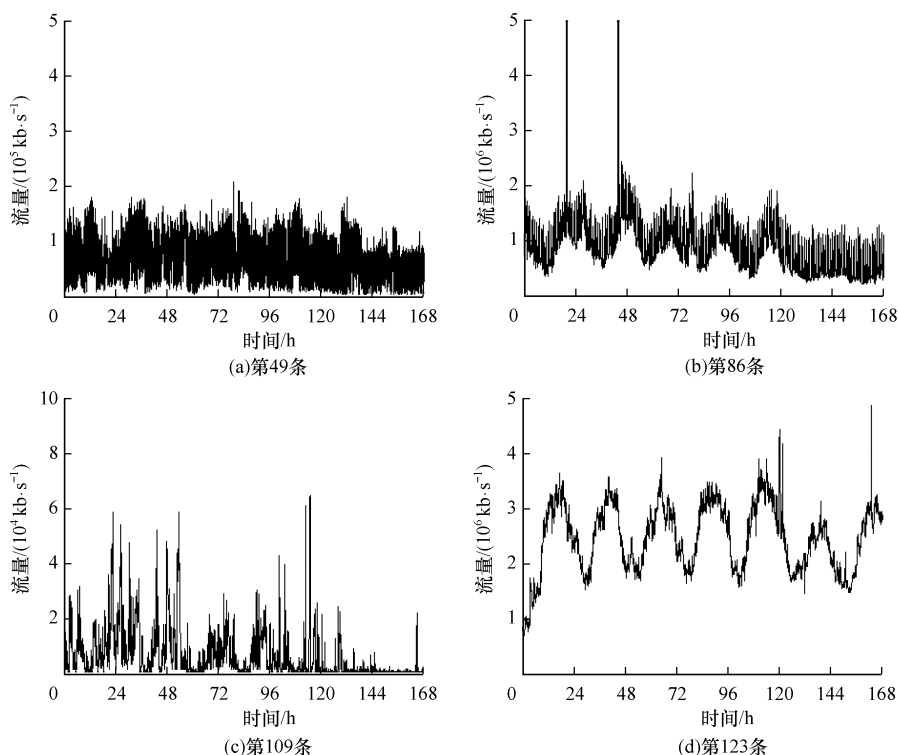


图 1 Abilene 网络部分 OD 流

表 1 Abilene 网部分 OD 流变化模式总结

OD 流	24 h 之内	24 h 与 24 h 之间
第 49 条	波动剧烈, 无明显峰值	趋势类似
第 86 条	波动剧烈, 有波峰波谷	趋势变化较大
第 109 条	波动剧烈, 有多个峰值	趋势不同
第 123 条	有明显波峰波谷	趋势类似

针对以上问题, 本文提出基于 OD 流聚类的自适应多 Elman 算法 (AMElman)。AMElman 的核心是 Elman 神经网络<sup>[13]</sup>, 其结构如图 2 所示。

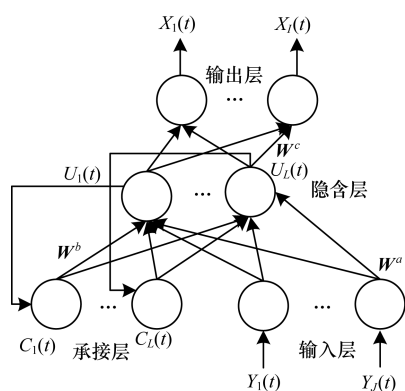


图 2 Elman 神经网络的结构

其中,  $Y(t) = (Y_1(t), Y_2(t), \dots, Y_J(t))$  为输入层向量;  $X(t) = (X_1(t), X_2(t), \dots, X_I(t))$  为输出层向量;  $U(t) = (U_1(t), U_2(t), \dots, U_L(t))$  为隐含层向量;  $C(t) = (C_1(t), C_2(t), \dots, C_L(t))$  为承接层向量。

$W^a, W^b, W^c$  分别为输入层到隐含层, 承接层到隐含层和隐含层到输出层的权值矩阵。Elman 的数学模型为:

$$\begin{cases} U(t) = f[W^a Y(t-1) + W^b C(t)] \\ C(t) = U(t-1) \\ X(t) = g[W^c U(t)] \end{cases}$$

其中, 隐含层的传递函数  $f(\cdot)$  为非线性函数, 输出层的传递函数  $g(\cdot)$  为线性函数。集合:

$$W = \{ (w_{j,m}^a, w_{n,m}^b, w_{m,i}^c) \mid j \in [1, J], m \in [1, L], n \in [1, L], i \in [1, I] \} \quad (2)$$

决定了映射关系  $F(\cdot)$ 。

相比于 BP 网络等静态神经网络, Elman 能够对历史输入进行记忆, 以更好映射出非线性和动态性特征。对混合 SDN 的 OD 流按照时间序列的变化模式进行聚类, 能够强化训练样本集合的关键特征, 得到针对相应特征的多个 Elman 神经网络。各 Elman 神经网络可并行训练, 且每个 Elman 神经网络的训练样本维度也相应降低, 从而提高整体的训练速度。而利用 SDN-OD 流可持续精确测量的特点, 能够动态追踪网络流量的变化, 及时调整多 Elman 神经网络的参数, 提高算法的自适应性。下面对 AMElman 算法进行详细介绍。

### 3 流量矩阵的自适应估计

AMElman 算法包括 3 个主要模块, 即多 Elman 网络训练、流量矩阵估计和参数自适应调整, 算法原

理如图 3 所示。

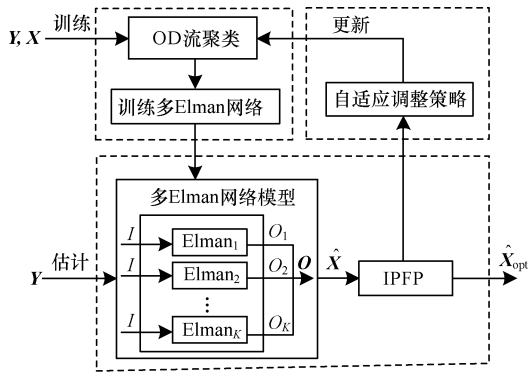


图 3 AMElman 原理

### 3.1 多 Elman 网络训练

在本模块中,首先利用  $K$  均值 ( $K$ -means) 算法<sup>[14]</sup>对训练样本的输出样本  $\{X(t)\}_{t=1}^S$  中的  $\{X_i(t)\}_{t=1}^S, i \in [1, I]$  按照时间序列变化模式的相似度进行聚类,得到反映不同变化模式关键特征的子样本集合  $\{\{X^k(t)\}_{t=1}^S | k \in [1, K]\}$ , 其中,  $K$  为聚类数。时间序列变化模式的相似度用欧氏距离:

$$\text{dist}[\{X_i(t)\}_{t=1}^S, \{X_j(t)\}_{t=1}^S] = \sqrt{\sum_{t=1}^S (X_i(t) - X_j(t))^2} \quad (3)$$

进行度量,使得聚类之后,不同类之间的 OD 流变化模式差别尽可能大,同一类内的 OD 流变化模式尽可能小。

然后以  $\{Y(t)\}_{t=1}^S$  作为输入,  $\{X^k(t)\}_{t=1}^S$  作为输出,训练相应的 Elman 神经网络,得到多 Elman 网络模型。模型中第  $k$  个 Elman 网络对应的权重集合为:

$$W^k = \{(w'_{j,m}, w'_{n,m}, w'_{m,i}) | j \in [1, J], m \in [1, L], n \in [1, L], i \in [1, I^k]\}$$

即将式 (2) 的单一权重集合  $W$  分解为多个权重集合  $Q = \{W^k | k \in [1, K]\}$ , 以反映不同变化模式的特征。

### 3.2 流量矩阵估计

在本模块中,利用 3.1 节得到多 Elman 网络模型,将链路流量  $Y$  作为输入,即可得到流量矩阵的估计值  $\hat{X}$ 。由于  $\hat{X}$  可能不满足式 (1) 的约束条件,因此还需要对其做进一步调整,以得到最终的估计值  $\hat{X}_{\text{opt}}$ 。本文采用迭代比例拟合 (Iterative Proportional Fitting Procedure, IPFP) 算法来进行调整。该算法在已知链路流量  $Y$  和路由矩阵  $A$  的前提下,通过对初始值  $\hat{X}$  在满足式 (1) 约束的条件下进行反复调整,最终得到满足误差要求的流量矩阵  $\hat{X}_{\text{opt}}$ 。

IPFP 的调整过程如下:

(1) 初始条件为多 Elman 网络的输出  $\hat{X}$  作为初始值;路由矩阵  $A$ ;链路流量  $Y$ 。

(2) 执行过程

**步骤 1** 设定最大迭代次数  $K$ , 收敛误差  $\varepsilon$ , 计算当前链路误差  $\text{err} = \|Y - A\hat{X}\| / \|Y\|$ 。

**步骤 2** 重复下列步骤,直到迭代次数大于  $K$  或者  $\text{err}$  小于  $\varepsilon$ :

for  $j = 1, 2, \dots, J$

$$\hat{Y} = A(j, :) * \hat{X}$$

$$\hat{X} = A(j, :)^T \times (Y_j / \hat{Y} * \hat{X}) + (\text{ones}(1, I) - A(j, :)^T) \times \hat{X}$$

$$- A(j, :)^T \times \hat{X}$$

end for

其中,  $A(j, :)$  为  $A$  的第  $j$  行;  $\text{ones}(1, I)$  为全 1 矩阵。

**步骤 3** 输出调整后的  $\hat{X}$ , 即  $\hat{X}_{\text{opt}}$ 。

### 3.3 参数自适应调整

本文的多 Elman 网络模型由多个不同参数的 Elman 网络组成。由于  $X_i(t)$  变化模式具有随机性, 为了适应这种变化,需对各个 Elman 网络的参数进行动态调整。在该模块中,设定 Elman 网络参数的有效期为  $T$ , 当参数过期时,对距离当前时刻最近的一段时期内的流量矩阵分量估计值进行  $K$ -means 聚类,结果与用于训练当前多 Elman 网络时流量矩阵分量的聚类结果进行对比。

对比结果分 3 种情况处理:(1)对于类内元素发生变化的分组,意味着当前的 Elman 网络参数已经不适用,故标记为“需重新测量”;(2)对于类内元素未发生变化的分组,但分组中所含 SDN-OD 流  $X_{\text{sdn}}$  不满足:

$$\text{dst}(X_{\text{sdn}}, Z_{C(X_{\text{sdn}})}) \leq \rho D_{C(X_{\text{sdn}})}, \rho \geq 0 \quad (4)$$

则标记为“需重新测量”。 $C(X_{\text{sdn}})$  为  $X_{\text{sdn}}$  所在分组的编号;  $Z_{C(X_{\text{sdn}})}$  为训练阶段该分组的中心;  $D_{C(X_{\text{sdn}})}$  为训练阶段分组半径。警戒参数  $\rho$  越小,越有助于提高整体的估计精度,但过小可能导致过多的分组被标记为“需重新测量”,过大则失去警戒意义;特别的,若分组中不含有 SDN-OD 流,则该分组直接标记为“需重新测量”。(3)除情况 (1)、情况 (2) 外,分组均标记为“无需重新测量”。所有标记为“需重新测量”的分组内的 OD 流在 3.1 节的模块中重新分组并训练相应的 Elman 网络,更新多 Elman 网络,使其适应新的网络状态。

## 4 算法描述

AMElman 算法中所涉及的变量说明如表 2 所示。

表 2 AMElman 算法中的变量

名称	含义
$TTB_i$	Train Time Bucket, 训练时段。在第 $i$ 个训练时段内对 $X, Y$ 进行测量, 得到用于训练神经网络的样本。 $TTB_i(t_b, t_e)$ 中, $t_b$ 表示 $TTB_i$ 的起始时刻, 记为 $TTB_i \cdot t_b$ ; $t_e$ 表示 $TTB_i$ 的终止时刻, 记为 $TTB_i \cdot t_e$ 。
$ETB_i$	Estimate Time Bucket, 估计时段。在第 $i$ 个估计时段内对流量矩阵 $X$ 进行估计。 $ETB_i(t_b, t_e)$ 中, $t_b$ 表示 $ETB_i$ 的起始时刻, 记为 $ETB_i \cdot t_b$ ; $t_e$ 表示 $ETB_i$ 的终止时刻, 记为 $ETB_i \cdot t_e$ 。
$T$	Elman 网络参数的有效期。当 $ETB_i = T$ 时, 判断当前多 Elman 网络的适应性, 并对 ElmanSet 做相应的更新。
$ElmanSet$	根据对流量矩阵 $X$ 分量的分组情况, 训练出相应的 Elman 神经网络, 构成集合 $ElmanSet = ElmanSet^M \cup ElmanSet^S$ 。其中, $ElmanSet^M$ 表示需要利用重新测量的 $Y$ 和 $X$ 来更新的 Elman 集合; $ElmanSet^S$ 表示保持不变的 Elman 集合。
$L$	$L = \{1, 2, \dots, I\}$ 为流量矩阵 $X$ 的分量的下标集合, $I$ 为 $X$ 的维度。
$K$	K-Means 中的聚类数。
$G, G^M, G^S$	$G = \{1, 2, \dots, K\}$ , $G^M \cup G^S = G$ , $G^S \cap G^M = \emptyset$ , 其中, $K$ 为分组数; $G$ 为流量矩阵 $X$ 的分量的分组编号集合; $G^M$ 为 $X$ 中需要重新测量的分量所在分组的编号集合; $G^S$ 为 $X$ 中不需要重新测量的分量所在分组的编号集合。
$X', X^M, X^S$	$X'$ 表示流量矩阵 $X$ 的所有 OD 流组成的集合。 $X' = X^M \cup X^S$ , $X^M = \bigcup_{k \in G^M, i \in L_k} X_i^k$ , $X^S = \bigcup_{k \in G^S, i \in L_k} X_i^k$ , 其中, $X^M$ 表示 $X$ 中需要重新测量的 OD 流集合; $X^S$ 表示 $X$ 中不需要重新测量的 OD 流集合。
$\pi, \pi^M, \pi^S$	将 $X$ 的分量划分为 $K$ 组, 对应于将集合 $L$ 进行 K-分划, 得到子集簇 $\pi = \{L_1, L_2, \dots, L_K\}$ 其中, $\pi$ 满足 ① $L = L_1 \cup L_2 \cup \dots \cup L_K$ ② 当 $i \neq j$ 时, $L_i \cap L_j = \emptyset$ 。另有, $\pi^M = \{L_k \mid k \in G^M\}$ , $\pi^S = \{L_k \mid k \in G^S\}$ , $\pi = \pi^M \cup \pi^S$ , $\pi^M \cap \pi^S = \emptyset$ , 其中, $\pi^M$ 表示 $X$ 中需要重新测量的 OD 流的下标集合; $\pi^S$ 表示 $X$ 中无需重新测量的 OD 流的下标集合。

AMElman 算法的完整步骤如下:

(1) 初始条件

$$TTB_i \cdot t_b = t_0, TTB_i \cdot t_e = TTB_i \cdot t_b + T_1$$

$$ETB_i \cdot t_b = ETB_i \cdot t_e = TTB_i \cdot t_e$$

其中,  $T_1$  为  $TTB_i$  的长度,  $i = 1$ 。

$$ElmanSet^M = \{Elman_i \mid i \in G^M\}$$

$$ElmanSet^S = \{Elman_i \mid i \in G^S\}$$

$$G^M = \{1, 2, \dots, K\}, G^S = \emptyset$$

$$X^M = X, X^S = \emptyset$$

(2) 执行过程

步骤 1 测量  $TTB_i$  内的  $Y$  和  $X^M$ , 对  $X^M$  内的元素利用 K-means 方法, 分为  $(K - |G^S|)$  组, 得到  $L$  的

$K^M$ -分划, 更新  $\pi^M$ , 得到  $\pi$ 。更新  $X^M$ 。

步骤 2 以  $TTB_i$  内的  $Y$  为输入,  $X^M$  为输出, 训练并更新  $ElmanSet^M$ 。

步骤 3 将  $ETB_i \cdot t_e$  时刻的  $Y$  分别作为  $ElmanSet^M$  和  $ElmanSet^S$  的输入, 得到估计的  $\hat{X}^M$  和  $\hat{X}^S$ , 合并后得到  $\hat{X}$ 。

步骤 4 对  $\hat{X}$  使用 IPFP 算法进行处理, 得到  $\hat{X}_{opt}$ 。若  $ETB_i \cdot t_e - ETB_i \cdot t_b < T$ , 则输出  $\hat{X}_{opt}$ , 更新  $ETB_i \cdot t_e = ETB_i \cdot t_e + 1$ , 转步骤 3。否则, 转步骤 5。

步骤 5 判断是否需要更新  $ElmanSet$  进行更新:

令  $G^S = G^M = \emptyset$ , 对  $\hat{X}_{opt}$  的分量利用 K-means 方法分为  $K$  组, 得到  $L$  的 K-分划, 对应的子集簇为  $\pi' = \{L'_1, L'_2, \dots, L'_K\}$ 。比较  $\pi'$  和  $\pi$ , 若  $L'_i \neq L_i$ , ( $i \in G$ ), 即第  $i$  个分组内的成员发生变化, 则  $G^M = G^M + \{i\}$ ; 否则, 若该组中含有持续测量的 OD 流, 且式 (4) 成立, 则  $G^M = G^M + \{i\}$ , 否则  $G^S = G^S + \{i\}$ 。更新  $\pi^M, \pi^S, ElmanSet^M$  和  $ElmanSet^S$ 。更新  $TTB_i$  和  $ETB_i$ , 转步骤 1。

## 5 算法有效性评估

为验证本文所提算法的有效性, 用 Abilene 网络的真实数据进行测试。数据取自 Zhang 在 Abilene 上采集的 2004 年 3 月 1 日 - 2004 年 3 月 7 日期间的真实流量, 采集频率为 5 min 一次。本文假设一种最初始的混合 SDN 场景, 即 Abilene 网络中仅有一个 SDN 节点 STTLng (西雅图), 其他节点为 IP 节点, 拓扑如图 4 所示。该拓扑中共有 30 条链路, 144 条 OD 流, 其中, SDN-OD 流为 23 条; IP-OD 流为 121 条。

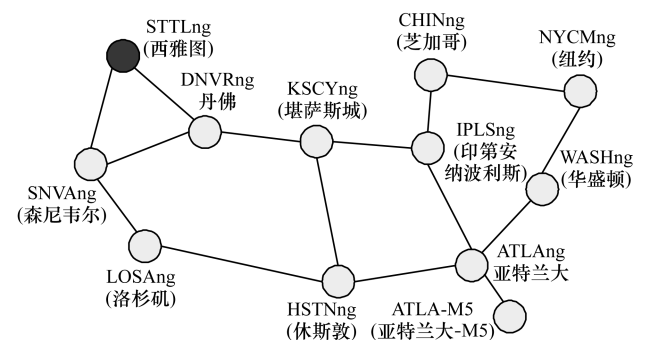


图 4 混合 SDN 的拓扑结构

由于广义层析重力 (General Tomography, GT) 模型是公认的一种精确的流量矩阵估计方法<sup>[12]</sup>, 因此本文以此来对比 AMElman 方法的估计效果。为消除 OD 流本身的大小对误差的影响, 定义式 (5) 所示的比例化均方误差 (scaled Mean Square Error, sMSE), 来评价流量矩阵的估计效果:

$$sMSE(\hat{X}, X) = \frac{\|\hat{X} - X\|^2}{\|X\|_1} = \frac{\sum_i (\hat{X}_i - X_i)^2}{\sum_i X_i} \quad (5)$$

### 5.1 算法参数设置

各 Elman 网络隐含层与承接层节点数相同,由经验公式  $S_i = \lceil \sqrt{M_i \times N_i} \rceil + C, i = 1, 2, \dots, K$  决定,其中,  $C \in [0, 10]$  且取  $C = 0$ ;  $K$  为 OD 流聚类数;  $M_i, N_i$  分别为第  $i$  个 Elman 网络的输入层节点数和输出层节点数。输入层节点数对应于链路条数,即  $M_i = 30, i = 1, 2, \dots, K$ ; 输出层节点数对应于  $K$  个聚类中每类的 OD 流个数,即  $N_i = |L_i|, i = 1, 2, \dots, K, L_i$  的定义参见表 2。隐含层传递函数为 TanSig, 输出层和承接层传递函数为 Purelin。

对警戒参数  $\rho$  和聚类数  $K$  的设定,需要综合考虑算法的估计精度和性能。 $\rho$  取值范围通常为  $[1, 2]$ 。经过多次试验,选择聚类数  $K = 6$  比较合适。

### 5.2 单 Elman 与多 Elman 对估计精度的影响

设 AMElman 算法的警戒参数  $\rho = 0$ , 即每个训练周期内强制对所有的 OD 流进行重新测量,防止由于  $\rho$  设置的不恰当而引入额外的估计误差。

图 5、图 6 是单 Elman 网络 (即  $K = 1$ ) 与多 Elman 网络 ( $K = 6$ ) 在 144 h 内的估计效果对比,包含 2 个训练与估计周期:  $TTB_1 = (0 \text{ h}, 24 \text{ h}), ETB_1 = (24 \text{ h}, 72 \text{ h}); TTB_2 = (72 \text{ h}, 96 \text{ h}), ETB_2 = (96 \text{ h}, 144 \text{ h})$ 。从图 5 中可以看出,单 Elman 网络 (即不分类,  $K = 1$ ) 的估计误差的平均值为 3.73%, 远高于 GT 的估计误差平均值 2.39%。而由图 6 可见,由于 Elman 网络通过分类训练,针对不同类型的 OD 流单独训练,使得估计模型更加准确,估计误差的平均值为 1.85%, 提高了整体的估计精度。

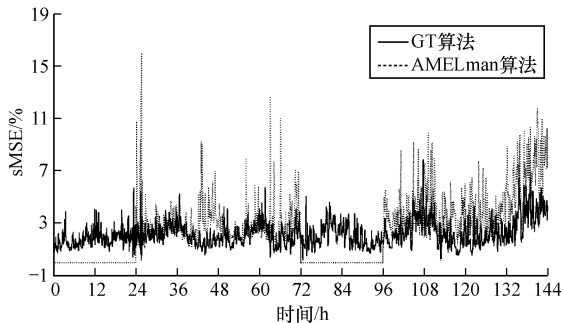


图 5 单 Elman 网络 ( $K = 1$ ) 的估计精度

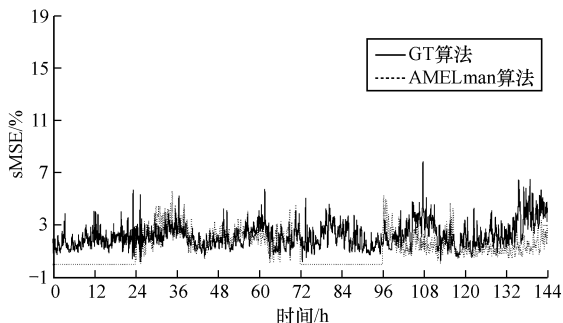


图 6 多 Elman 网络 ( $K = 6$ ) 的估计精度

### 5.3 TTB 对估计精度的影响

图 7 ~ 图 9 为不同 TTB 对估计精度的影响,  $ETB_i$  设为 48 h。  $TTB_i$  越大, 表明训练多 Elman 网络的样本数越大, 必然导致神经网络的训练速度变慢。

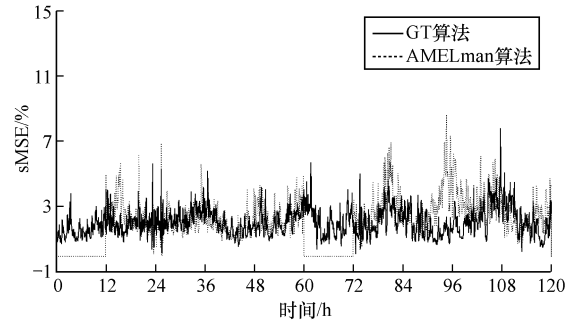


图 7 TTB 为 12 h 时 AMElman 的估计精度

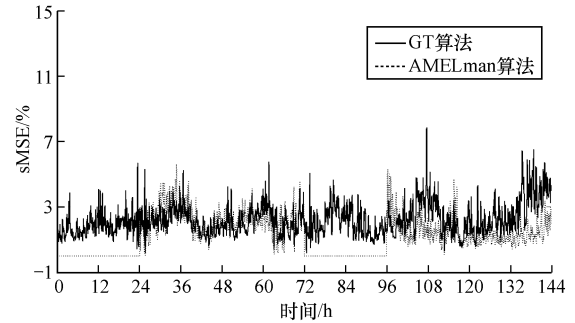


图 8 TTB 为 24 h 时 AMElman 的估计精度

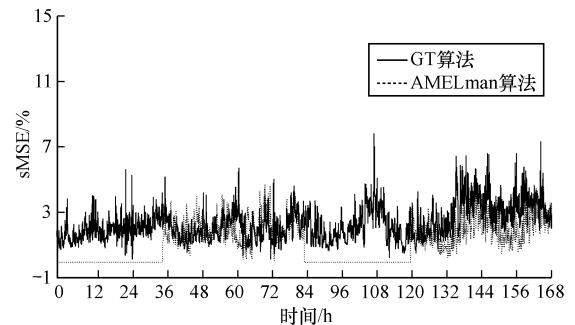


图 9 TTB 为 36 h 时 AMElman 的估计精度

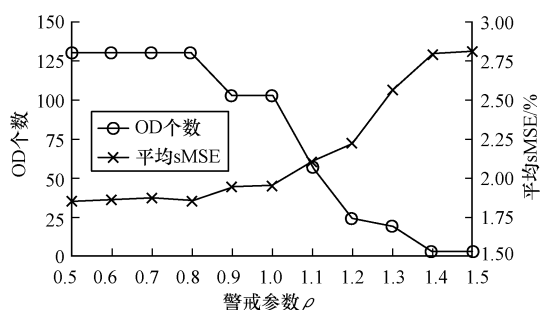
实验表明,虽然可以通过减小  $TTB_i$  来提高训练速度,但过小的  $TTB_i$  会导致估计精度降低,如图 7 所示,  $TTB_i$  为 12 h, 平均 sMSE 为 2.60%。但并非 TTB 越大,估计精度就越高。图 9 的  $TTB_i$  为 36 h, 平均 sMSE 为 2.07%, 低于图 8 所示的情况,  $TTB_i$  为 24 h, 平均 sMSE 为 1.85%。因此,要结合网络流量的实际变化情况选择合适的  $TTB_i$ 。由于网络流量一般表现出明显的日周期性,通常选择  $TTB_i = 24 \text{ h}$  较为合适。表 3 所示为 AMElman 与 GT 的估计精度比较。可见, AMElman 算法在选择了合适的  $TTB_i$  后,估计精度明显优于 GT。

表 3 AMElman 与 GT 的估计精度比较 %

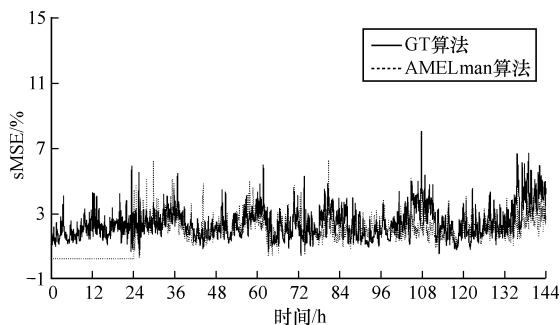
评价指标	AMElman 算法			GT 算法
	TTB 为 12 h	TTB 为 24 h	TTB 为 36 h	
最小 sMSE	0.02	0.02	0.09	0.17
最大 sMSE	8.69	5.60	4.80	7.84
平均 sMSE	2.60	1.85	2.07	2.39

#### 5.4 警戒参数 $\rho$ 对自适应性和估计精度的影响

为分析警戒参数  $\rho$  对自适应性和估计精度的影响,选取第 2 个训练与估计周期  $TTB_2 = (72 \text{ h}, 96 \text{ h})$ ,  $ETB_2 = (96 \text{ h}, 144 \text{ h})$  的数据进行分析。图 10 为警戒参数  $\rho$  取不同值时所需重新测量的 OD 数及对应的估计精度。

图 10 警戒参数  $\rho$  对自适应开销和估计精度的影响

可见,在一定范围内 ( $0.8 \leq \rho \leq 1.4$ ),随着  $\rho$  的取值逐渐增大,所需重新测量的 OD 数下降,多 Elman 网络的适应性调整的开销随之降低,但同时会导致估计误差增大。因此,设定一个合理的  $\rho$  能够保证估计误差在可接受范围内降低自适应调整的开销。由式(4)可知,判断某分组内的 OD 流是否可以继续沿用当前的 Elman 网络进行估计,需衡量该分组内 OD 流与用来训练当前 Elman 网络的 OD 流之间的相似度。 $\rho$  反映了相似度变化程度的可接受上限,适当放大该上限,能够防止因 OD 流的随机波动而导致 Elman 网络进行重新训练。图 11 所示为相似度变化程度上限为 20%,即  $\rho = 1.2$  时,AMElman 的估计效果。此时仅需重新测量 24 条 OD 流(其中,5 条为 SDN-OD 流),即能达到优于 GT 的估计效果。

图 11  $\rho$  为 1.2 时 AMElman 的估计精度

#### 5.5 AMElman 综合性能分析

由 5.2 节 ~ 5.4 节的分析可知,影响 AMElman 算法性能的关键因素有 3 个:OD 流聚类数,训练时段的长短以及警戒参数。这 3 个参数相互独立,可分别调优。

OD 流聚类数对应多 Elman 网络的个数,表 4 所示为训练单 Elman 网络 ( $K = 1$ ) 与多 Elman 网络 ( $K = 6$ ) 分别所需的 CPU 时间。训练时段为  $TTB = (0 \text{ h}, 24 \text{ h})$ ,CPU 频率为 2.5 GHz。可见,对 OD 流进行合理的聚类,能够提高算法的估计精度,但算法开销也会增大。由于每个 Elman 网络可独立进行训练,因此 AMElman 更适合采用并行方法进行训练。

表 4 聚类数对算法性能的影响

K	Elman 网络输出层节点数	CPU 时间/s	平均 sMSE/%
1	144	11.700 1	3.77
	50	9.406 9	
	44	6.676 8	
	3	1.716 0	
6	7	1.809 6	1.82
	24	2.012 4	
	16	1.825 2	

从表 3 可看出,训练时段增大,算法的估计精度不一定降低,但由于训练 Elman 网络的样本数度增加,导致算法的运算开销增加。

从图 10 可看出,警戒参数增大,算法的估计精度降低,但由于需要重新测量的 OD 流减少,相应地需要重新训练的 Elman 网络数降低,使算法的运算开销降低。

综上所述,通过确定合理的 OD 流聚类数、训练时段以及警戒参数,AMElman 算法能够在不引入过大的运算开销下,达到预期的估计效果。特别地,采用并行计算能够更加体现出 AMElman 算法的优势。

## 6 结束语

本文研究混合 SDN 的流量矩阵估计问题,提出基于 OD 流聚类的自适应多 Elman 网络估计算法。通过对 OD 流按时间变化模式进行聚类,强化了训练样本的关键特征,降低了训练样本的维度,提高了算法估计的准确性。同时,利用 SDN-OD 流追踪混合 SDN 的变化,动态调整算法参数,提高了算法的自适应性。实验结果表明,与传统算法相比,该算法具有更低的估计误差和更好的自适应性。

混合 SDN 中 SDN 节点一般采用增量式部署方案。SDN 节点的增加,为直接测量更多的 OD 流提供了方便,有利于更准确地跟踪网络状态的变化。而合理选择 SDN 节点的部署,能够使 SDN OD 流在

OD 流各聚类中分布更均匀,有利于更有效地判断相应 Elman 网络的适用性。因此,下一步的工作重点是研究混合 SDN 中 SDN 节点的部署比例和位置对流量矩阵估计方法的影响。

### 参考文献

- [1] Lin Pingping, Hart J, Krishnaswamy U, et al. Seamless Interworking of SDN and IP[J]. ACM Sigcomm Computer Communication Review, 2013, 43(4): 475-476.
- [2] Jain S, Kumar A, Mandal S, et al. B4: Experience with a Globally-deployed Software Defined WAN [J]. ACM Sigcomm Computer Communication Review, 2013, 43(4): 3-14.
- [3] Guo Yingya, Wang Zhiliang, Xia Yin, et al. Traffic Engineering in SDN/OSPF Hybrid Network [C]// Proceedings of IEEE the 22nd International Conference on Network Protocols. Washington D. C., USA: IEEE Press, 2014: 563-568.
- [4] Vardi Y. Network Tomography: Estimating Source-destination Traffic Intensities from Link Data [J]. Journal of the American Statistical Association, 1996, 91(433): 365-377.
- [5] Cao Jin, Davis D, Vander W S, et al. Time-varying Network Tomography: Router Link Data [J]. Journal of the American Statistical Association, 2000, 95(452): 1063-1075.
- [6] Zhang Yin, Roughan M, Duffield N, et al. Fast Accurate Computation of Large-scale IP Traffic Matrices from Link Loads [J]. ACM SIGMETRICS Performance Evaluation Review, 2003, 31(1): 206-217.
- [7] Nucci A, Cruz R, Taft N, et al. Design of IGP Link Weight Changes for Estimation of Traffic Matrices [C]// Proceedings of the 23rd Annual Joint Conference of IEEE Computer and Communications Societies. Washington D. C., USA: IEEE Press, 2004: 2341-2351.
- [8] Papagiannaki K, Taft N, Lakhina A. A Distributed Approach to Measure IP Traffic Matrices [C]// Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement. New York, USA: ACM Press, 2004: 161-174.
- [9] Lakhina A, Papagiannaki K, Crovella M, et al. Structural Analysis of Network Traffic Flows [C]// Proceedings of Joint International Conference on Measurement and Modeling of Computer Systems. New York, USA: ACM Press, 2004: 61-72.
- [10] 卢学强, 梁雪慧, 卢学军. 神经网络方法及其在非线性时间序列预测中的应用 [J]. 系统工程理论与实践, 1997, 17(6): 98-100.
- [11] 张大方, 王晓阳. 基于多神经网络的流量矩阵估计方法: 中国, CN102136087A [P]. 2011-07-27.
- [12] 蒋定德, 王兴伟, 郭磊, 等. 大尺度 IP 骨干网络流量矩阵估计方法研究 [J]. 电子学报, 2011, 39(4): 763-771.
- [13] 张立军, 苑迪. 基于 GA-Elman 动态回归神经网络的股价预测模型研究 [J]. 华东经济管理, 2008, 22(9): 79-82.
- [14] MacQueen J. Some Methods for Classification and Analysis of Multivariate Observations [C]// Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability. Berkeley, USA: University of California Press, 1967: 281-297.
- [5] Tauchim I, Murata Y. Ad Hoc Routing Protocol Avoiding Route Break Based on AODV [C]// Proceedings of the 38th Annual Hawaii International Conference on System Sciences. Washington D. C., USA: IEEE Computer Society, 2005: 322-329.
- [6] Perkins C, Bhagwat P. Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers [J]. ACM SIGCOMM Computer Communication Review, 1994, 24(4): 234-244.
- [7] Hu Y C, Johnson D B, Perrig A. SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks [J]. Ad Hoc Networks, 2003, 1(1): 175-192.
- [8] 杨玉华, 刘培宁, 刘际伟. NS-2 的仿真模拟技术分析 [J]. 计算机工程, 2005, 31(15): 110-112.
- [9] 高明亮, 曹晟熙. 移动自组织网络 AODV 路由协议的仿真与性能分析 [J]. 控制工程, 2009, 16(S1): 142-144.
- [10] 张天明, 王培康. Ad Hoc 移动网络 AODV 协议的分析与改进 [J]. 计算机辅助工程, 2008, 17(3): 61-64.
- [11] 任智, 郭伟, 苏静, 等. 基于跨层协同设计的高效 AODV 改进路由算法 [J]. 计算机学报, 2007, 30(5): 838-843.
- [12] 李庆, 刘聪, 江汉红, 等. Ad Hoc 网络中 AODV 路由协议的优化 [J]. 计算机工程, 2008, 34(13): 107-109.
- [13] 刘俊, 徐昌彪. 基于 NS2 的网络仿真探讨 [J]. 计算机应用研究, 2002, 22(9): 75-78.
- [14] 符琦, 蒋碧霞, 蒋瑞林. 基于 NS2 的 Ad Hoc 网络路由协议的模拟实现 [J]. 计算机工程与应用, 2006, 43(5): 149-152.
- [15] 寇小文, 周铁军. NS 网络仿真技术及其结果输出和分析 [J]. 信息技术, 2007, (12): 90-92, 96.
- [16] 花晓菲. 基于 NS2 的 Ad hoc 网络路由协议性能分析比较 [J]. 无线通信技术, 2006, 5(3): 29-33.

编辑 刘冰

编辑 刘冰

(上接第 102 页)