

顶层是一个标准的前馈网络,利用 softmax 输出来对消息的爆发进行预测.下面结合图 3 详细介绍这 3 个部分的作用和计算方法.

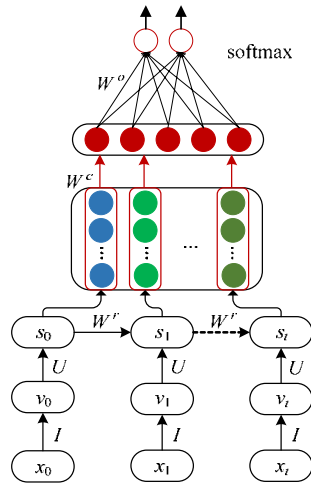


Fig.3 Architecture of SMOP
图 3 SMOP 模型架构

3.1.1 底层网络

底层网络采用 RNN. RNN 独特的循环网络结构使其能够保持和传递内部的状态信息,建模动态的时间序列行为.与前馈神经网络相比, RNN 能够处理任意的输入序列长度,建模多时间尺度的序列关系,十分适合于学习和抽取序列数据中的特征和模式.底层从上到下又分为 3 层.

- 第 1 层是输入层(input layer), x_t 是 RNN 第 t 个时间步骤的输入值,对应输入序列 X 中的第 t 个元素,它可以只包含时刻信息,也可以同时包含转发用户特征等其他信息.这里将时刻和用户等输入映射成向量的形式输入到上层网络中进行计算,映射的方式在下文中会详细说明.
- 第 2 层是嵌入层(embedding layer), I 为索引操作, v_t 为 x_t 经过索引操作后得到输入的向量形式, U 为输入权重矩阵.
- 第 3 层是 RNN 的隐藏层, W^r 为循环权重矩阵, s_t 为 RNN 第 t 个时间步骤隐藏层的输出,其计算公式如公式(2)、公式(3)所示.

$$v_t = I(x_t) \tag{2}$$

$$s_t = f(Uv_t + W^r s_{t-1}) \tag{3}$$

3.1.2 中间层网络

中间层网络的功能是对 RNN 的隐藏层输出进行非线性变化,得到消息传播过程的速率函数 $\lambda(t)$. 本文将社交网络上消息的转发过程建模成随机点过程,设某条消息转发时刻 t_i 为顺序产生的非负的随机变量,则称 t_i 为正实数域上的一个点过程.点过程的速率函数 $\lambda(t)$ 可以唯一刻画一个点过程的所有特征,其定义如公式(4)所示.

$$\lambda(t) = \lim_{\Delta t \rightarrow 0} \left[\frac{N(t + \Delta t) - N(t)}{\Delta t} \middle| H_t \right] \tag{4}$$

其中, H_t 表示时刻 t 之前的历史过程. $\lambda(t)$ 反映了点过程速率随时间的变化关系,完全刻画了一个点过程的特征.例如,对于 Poisson 过程来讲,其 $\lambda(t)$ 是一个不随时间变化的常量 λ ,即在 Poisson 过程中,下一个事件发生的时刻与历史的事件 H_t 没有依赖关系,事件之间的时间间隔服从参数为 λ 的指数分布.对于社交网络的消息传播过程,前人的研究表明,当前的转发受到历史过程 H_t 的影响,即其转发序列存在长距离的依赖关联.为了刻画这种依赖关系,先前的研究定义了各种各样的速率特征函数.与手动定义速率特征函数的方法不同, SMOP 通过 RNN 网

络自动学习点过程的速率函数 $\lambda(t)$,定义如公式(5)所示.

$$\lambda(t)=\exp(W^c s_t+w(t-t_i)+b^c) \tag{5}$$

其中, W^c, w, b^c 为参数, s_t 为 RNN 当前时间步骤的隐藏层输出, $W^c s_t$ 表示了历史过程对 $\lambda(t)$ 的影响, t_i 为当前时刻的网络输入, $w(t-t_i)$ 表示了当前的转发动作对 $\lambda(t)$ 的影响,加 $\exp(\cdot)$ 是为了满足 $\lambda(t) > 0$ 的约束.在 SMOP 中,只需知道每个转发时刻的瞬时速率,即 $\lambda(t_i)$,如公式(6)所示.

$$\lambda(t_i)=\exp(W^c s_t+b^c) \tag{6}$$

因此,经过中间层网络学习,得到了消息的序列特征 $(\lambda(t_1), \lambda(t_2), \dots, \lambda(t_i))$,这些特征通过模型顶层的前馈网络进行处理后进行消息的分类. W^o 为顶层网络的权重矩阵,其计算公式如公式(7)所示.

$$\hat{y} = \text{softmax}(W^o s_t^T) \tag{7}$$

其中, \hat{y} 为模型最后的预测输出.

3.2 网络输入

由第 2 节的问题定义可知,模型的输入为每次转发动作发生的时间 t_i 和用户 u_i .在本模型中,参与网络计算的不是简单的实数值或是用户标识,而是包含丰富语义的嵌入式表示.如图 4 所示,输入首先会按照一定的规则映射成向量的形式,之后再进行下一步的计算.

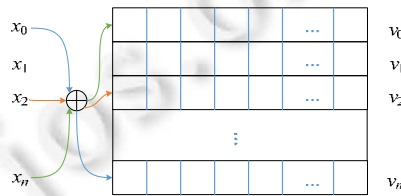


Fig.4 Indexing from input to embedding

图 4 输入映射成向量

需要指出的是,每条消息在观测时间 t_i 内的转发序列长度是不同的.为使最后进入前馈网络的特征具有相同的长度,我们对输入序列 X 做如下处理:首先,设置输入序列 X 的最大长度 L ,对于长度大于 L 的序列,采用从序列尾部的方式截断处理;对于长度小于 L 的序列,采用 padding 的方式,在序列的头部填充 0.

3.2.1 时刻向量表示

时间信息如何以一种神经网络易于理解和训练的方式输入,是一个值得探讨的问题:一方面,在消息的传播序列中,由于时间的自增长特点,转发时刻的取值范围非常大,如果直接以时刻作为实数值输入,会导致神经网络收敛慢,特别难训练;另一方面,时间天生具有周期性的特点,如年、月、周、日、时、分、秒等,而简单的实数值无法包含时间丰富的周期信息,因此在本文中,将时间的实数值映射到时间的周期中,嵌入成时间的向量表示.具体来讲,对于每一个时间周期,按照其上一级时间周期转换成该周期的长度,设置其在时间向量中的长度.以时间周期秒为例,其上一级时间周期分转化为周期秒的长度为 60,因此在时间向量中,对周期秒就设置 60 位的长度.对于某个时刻 t ,将其转化为周期秒表示后,对得到的周期长度取模得到 k ,则在时间向量中将相应周期的第 k 位设置为 1,其余位置 0,就得到时刻 t 在周期秒上的表示.其他时间周期采用类似的操作方法,如图 5 所示.

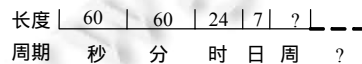


Fig.5 Time embedding

图 5 时间向量生成

3.2.2 用户向量表示

社交网络中的用户是消息的传播者和源动力,用户的兴趣和偏好对消息的转发起着关键作用.然而在现有

的点过程模型中,绝大部分只建模了消息的转发时间序列,没有深入挖掘和利用用户在传播中的作用.因此,为了建模用户兴趣和偏好在消息传播中的重要作用,本节将讨论如何对用户画像,将用户的特征表示成用户兴趣向量的形式.

如图 6 所示,第 1 步,我们将社交网络中用户的历史发布消息聚合成用户的代表文档,所有用户的代表文档构成用户的文档集合;第 2 步,在用户文档集合上,采用 LDA(latent dirichlet allocation)主题模型学习出每个用户代表文档的主题分布向量,主题分布向量的每一维表示了用户代表文档属于该主题的概率,概率越大,表示用户在历史上发布的涉及该主题的消息越多,即用户可能对该主题越感兴趣.因此,我们将 LDA 主题模型生成的用户代表文档的主题分布向量作为用户的向量表示.

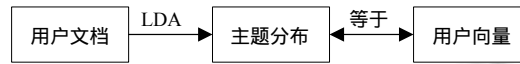


Fig.6 User embedding

图 6 用户向量生成

在基于循环神经网络的点过程中,在每一个输入步骤,将当前时刻映射成时间向量表示,将当前正进行转发的用户映射成用户兴趣向量表示,两个向量表示前后拼接起来作为一个整体,输入到循环神经网络的隐藏层进行计算.

3.3 基于门的循环网络单元

为了缓解 RNN 在训练过程中梯度消失(gradient vanishing)的问题,人们从参数初始化方式、激活函数的选择、网络单元的结构多个方面进行了研究.在 SMOP 模型中,主要采用基于门的循环网络单元来更好地训练模型,下面从网络结构的角度介绍目前常用的 LSTM^[10,18]和 GRU^[11]单元.

LSTM 利用门机制来解决梯度消失的问题,其内部结构如图 7 所示.它由 3 个控制门(gate)和 1 个内部存储单元(cell)组成,gate 是一种让信息选择性通过的机制,全 0 表示不让任何信息通过,全 1 表示让所有信息通过,cell 则起到了保持和传递信息的作用.3 个控制门依次是输入门(input gate, i_t)、遗忘门(forget gate, f_t)和输出门(output gate, o_t), g, h 为 $\tanh(\cdot)$ 激活函数, σ 为 $\text{sigmoid}(\cdot)$ 激活函数, x_t, c_t 和 s_t 分别是 LSTM 单元步骤 t 时的输入向量、内部状态向量和输出向量, z_t 就是标准 RNN 中的输出.为了形式化 LSTM 的前向计算过程,我们用带下标的 U, W 和 b 分别表示相应的门输入权重矩阵、循环权重矩阵和偏置向量.例如, U_i, W_i 和 b_i 分别表示输入门的输入权重矩阵、循环权重矩阵和偏置矩阵. LSTM 层的前向计算过程如下.

$$i_t = \sigma(U_i v_t + W_i s_{t-1} + b_i) \quad (8)$$

$$f_t = \sigma(U_f v_t + W_f s_{t-1} + b_f) \quad (9)$$

$$o_t = \sigma(U_o v_t + W_o s_{t-1} + b_o) \quad (10)$$

$$z_t = g(U_z v_t + W_z s_{t-1} + b_z) \quad (11)$$

$$c_t = i_t \circ z_t + f_t \circ c_{t-1} \quad (12)$$

$$s_t = o_t \circ h(c_t) \quad (13)$$

其中,符号 \circ 表示数组对应位置的元素相乘.

与 LSTM 相比,GRU 单元只有两个门:重置门(reset gate)和更新门(update gate),且没有内部记忆单元 cell,其结构如图 8 所示. GRU 层的前向计算过程如下.

$$r_t = \sigma(U_r v_t + W_r s_{t-1} + b_r) \quad (14)$$

$$z_t = g(U_z v_t + W_z s_{t-1} + b_z) \quad (15)$$

$$h_t = g(U_h v_t + W_h (s_{t-1} \circ r_t) + b_h) \quad (16)$$

$$s_t = (1 - z_t) \circ h_t + z_t \circ s_{t-1} \quad (17)$$

Greff 等人^[19]的研究结果表明,两者的性能差不多,提高性能的关键在于参数训练的过程和数据的规模.但

是,如果数据比较充足,建议选择 LSTM.

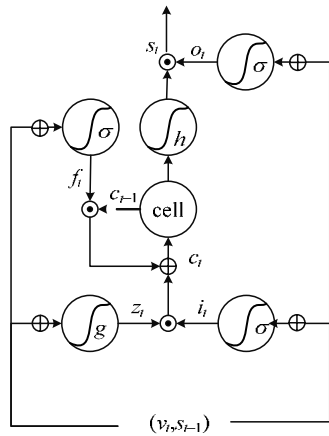


Fig.7 LSTM architecture
图 7 LSTM 单元结构

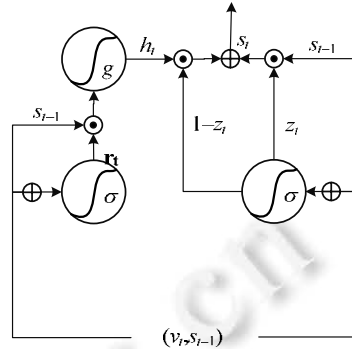


Fig.8 GRU architecture
图 8 GRU 单元结构

3.4 模型训练

模型的训练采用 mini-batch 随机梯度下降,参数更新采用 Adadelta^[20]规则,mini-batch 的大小设置为 32.随机选取 80%的数据作为训练集,20%的数据作为测试集.在训练数据中,随机选取 10%的数据作为验证集.所有参数的初始化都采样自单变量的高斯分布,均值为 0,方差为 1,并且采用奇异值分解(singular value decomposition,简称 SVD)进行正交化.在网络的倒数第 2 层,引入 Dropout^[21]的方式避免过拟合.

4 实验评估

在本节中,我们通过真实的 Twitter 网络和新浪微博网络数据对 SMOP 模型进行评估.首先,我们将详细介绍实验数据的预处理及其相关特征;然后介绍相关的比较基准算法和评估指标;最后给出实验结果.

4.1 数据集

Twitter 数据集采用的是文献[12]公开的数据集,采集的时间是 2011 年 10 月 7 日~2011 年 12 月 31 日,原始的数据集包含的消息和转发消息数超过 32 亿条,数据处理的方式和文献[12]相同.处理后,前 15 天的微博数量为 166 076 个,将前 15 天又分成两个阶段:前 7 天用于训练,后 8 天用于测试,测试的样本数为 94 256.将剩余 14 天的时间用于消息的传播,即 t_r 为 14 天.

Weibo 数据集来自实验室自采的数据集,该数据包含了首发时间在 2016 年 10 月 10 日当天、持续时间至 2016 年 10 月 24 日的微博消息的完整转发关系, t_r 也为 14 天.

为了控制数据集的大小,我们保留了转发数在[50,50000]区间的消息进行研究,总计 10 000 条微博消息.

不失一般性,我们将微博在 t_r 时刻被转发的数量进行排序,选取转发数 top 5%以内的消息作为爆发消息,即正例;同时,为了提高训练样本的区分度,在正例和负例之间的转发数设置一个间隔,本文的间隔的取值为正例中消息最小转发数的 10%.经过处理后,用于实验测试的数据如下:对于 Twitter 数据集,爆发消息 1 801 条,最小转发数为 1 000 条,转发数间隔取 100.为了避免不平衡数据对算法性能带来的影响,在负例中随机选取了相同数目的消息作为负例.类似地,对于微博消息,爆发消息为 491 条,最小转发数为 750,转发数间隔取 75.

4.2 评估标准

社交网络消息的爆发检测问题定义为一个二元分类问题.因此,本文采用精确率(precision,简称 P)、召回率(recall,简称 R)和 F1 分数来对本文设计的模型和比较方法进行评估.设 S 表示测试样本集合,结合第 2 节中的符

号定义,精确率、召回率和 $F1$ 的定义如下.

$$P = \frac{\sum_{i \in S} y_i \cdot \hat{y}_i}{\sum_{i \in S} \hat{y}_i} \quad (18)$$

$$R = \frac{\sum_{i \in S} y_i \cdot \hat{y}_i}{\sum_{i \in S} y_i} \quad (19)$$

$$F1 = \frac{2 \cdot P \cdot R}{P + R} \quad (20)$$

4.3 基准方法

本文采用的比较基准为当前主流的爆发预测方法和流行度预测算法,主要分为 3 类:基于分类模型的方法、基于回归分析的模型和基于随机点过程的方法.

- 基于分类模型的方法

OSLOR(orthogonal sparse logistic regression)^[3]. 该模型是在逻辑回归方法的基础上演化而来,通过添加 L1 正则和正交正则达到联合优化社交网络关键节点选择和爆发预测的目的.模型定义如下:

$$h(X_i^t) = \frac{1}{1 + \exp(-\theta_0 - X_i^t \theta)} \quad (21)$$

其中, θ 为参数矩阵, X^t 为消息传播的状态矩阵.如 X_i^t 表示第 i 条消息在时刻 t 的状态,优化的目标是

$$F(\theta) = -\log L(\theta) + \frac{\beta}{4} \sum_{i,j} (\theta_i X_i^T X_j \theta_j)^2 + \gamma \|\theta\|_1 \quad (22)$$

其中, $L(\theta)$ 是 $h(X_i^t)$ 的极大似然函数.

- 基于回归分析的模型

LR(univariate linear regression)^[13]. LR 模型基于这样的观察,即消息的流行度(引用数、转发数等)经过 log 转换后,与该消息早期的流行度存在较强的相关性.LR 的定义如下:

$$\ln N_p(t_r) = \beta \ln N_p(t_i) + \xi \quad (23)$$

其中, $N_p(t_r)$ 为消息 p 在 t_r 时刻的流行度, $N_p(t_i)$ 为消息 p 在 t_i 时刻的流行度, ξ 表示高斯噪声.

MLR(Multivariate linear regression)^[5]. 该模型定义为

$$N_p(t_r) = \Theta \cdot X_p(t_i) \quad (24)$$

其中, $\Theta = (\theta_1, \theta_2, \dots, \theta_n)$ 是模型的参数向量, $X_p(t_i)$ 为模型的特征向量,定义如下:

$$X_p(t_i) = (x_p^1(t_i), x_p^2(t_i), \dots, x_p^n(t_i))^T \quad (25)$$

其中,将消息的初始转发时间至 t_i 分成 n 等分, $x_p^k(t_i)$ 为第 k 个时间间隔的转发数.与 LR 模型相比,MLR 对不同的时间段消息的流行度赋予不同的权重,对消息的早期流行度进行了更加细致的区分.

MRBF(MLR model with radial basis functions)^[5]. MRBF 为 MLR 模型的变种,其定义如下:

$$N_p(t_r) = \Theta \cdot X_p(t_i) + \sum_{p_c \in C} \omega_{p_c} \cdot RBF_{p_c}(p) \quad (26)$$

其中, C 为训练集中随机选取的消息集合, $p_c \in C$, $RBF_{p_c}(p)$ 为消息 p 的高斯 RBF(radial basis function)特征,其定义如下:

$$RBF_{p_c}(p) = \exp\left(-\frac{\|X(p) - X(p_c)\|^2}{2\sigma^2}\right) \quad (27)$$

其中, $X(p)$ 表示消息 p 的特征, RBF 特征用来捕获特定类型消息的特征模式.

- 基于点过程的模型

SEISMIC(self-exciting model of information cascades)^[12]. 该模型将消息的传播过程建模成一个双随机的自激励点过程(doubly stochastic self-exciting point process),其速率表达式如下:

$$\lambda_d(t) = p(t) \sum_{t_i < t} n_i \phi(t - t_i) \tag{28}$$

其中, $p(t)$ 表示消息的传染性, 表示用户转发消息 d 的概率; $\phi(t)$ 表示用户看到一条消息后, 他转发该消息的时延; n_i 表示用户 i 的粉丝数, 反映了网络结构的信息. 与传统的随机过程不同, $p(t)$ 自身也是一个随机过程.

4.4 实验结果

4.4.1 消息爆发预测

本节中, 我们通过详尽的实验来验证 SMOP 模型相对于同类的方法在消息爆发预测性能上的效果. 首先, 表 1 展示了在可观测消息传播过程为初始 1h, 输入数据只有转发时间序列, 而没有转发用户序列的情况下, 各种模型的效果比较.

Table 1 Performance evaluation on Twitter and Sina Weibo data
(observation time: 1h, input: time embedding)

表 1 在 Twitter 和新浪微博数据集上的算法性能评估(观测时长: 1h, 输入: 时间向量)

Method	Twitter			Weibo		
	Precision	Recall	F1	Precision	Recall	F1
LR	0.854 067	0.877 738	0.865 9	0.542 373	0.941 176	0.688 172
MLR	0.978 947	0.717 224	0.827 893	0.964 286	0.786 408	0.866 310
MRBF	0.978 947	0.717 224	0.827 893	0.957 734	0.789 320	0.865 366
SEISMIC	0.518 024	0.997 429	0.681 898	-	-	-
OSLOR	-	-	-	0.512 821	1.0	0.677 966
SMOP(GRU)	0.915 760	0.851 010	0.882 198	0.851 064	0.941 176	0.893 854
SMOP(LSTM)	0.839 673	0.930 722	0.882 857	0.895 833	0.86	0.877 551

由表 1 可知, SMOP 取得了目前所有方法中最优和次优的性能. 原始的 SMOP(RNN) 由于没有引入门的机制, 效果比 SMOP(GRU) 和 SMOP(LSTM) 差太多, 没有在表 1 中列出. SMOP(GRU) 和 SMOP(LSTM) 的性能相当, 在两个数据集上没有一致的优劣. 来看一下其他对比的方法: LR 虽然在 Twitter 数据上表现不错, 但在两个数据集上没有一致的性能表现, MLR 和 MRBF 在两个数据集上的 F1 得分都达到了 0.8 以上, 性能比其他比较方法稳定; 对于 SEISMIC, 由于其需要假设用户转发的响应时间服从幂律分布, 而在微博数据集上, 经测试, 用户的转发反映时间不符合幂律分布, 而更符合对数正态分布(logarithmic normal distribution), 因此, 我们只在 SEISMIC 方法原始的 Twitter 数据集上进行了测试, 实验中的各种参数都是文献[12]中的参数. 可以看出, SEISMIC 方法的 F1 只有 0.68, 得分较低. 对于 OSLOR 方法, 由于其需要用户的标识, 而 Twitter 数据没有提供, 因此只在微博数据机上做了实验, 其 F1 值小于 0.68, 表现同样不理想. 可以看到, 以 SEISMIC 为代表的基于点随机过程的方法, 由于其模型过于依赖样本数据的统计特征, 不能方便地适配到不同的数据环境中, 影响了实际的使用效果; 以 OSLOR 为代表的分类方法没有考虑到消息传播过程的序列关系, 只是将用户当作孤立的特征使用, 效果也不好.

表 2 展示了在可观测消息传播过程为初始 1h, 输入数据中同时加入时间和用户序列的效果.

Table 2 Performance evaluation on Sina Weibo data with user embedding
(observation time: 1h, input: time and user embedding)

表 2 在带有用户向量的新浪微博数据集上的算法性能评估(观测时长: 1h, 输入: 时间和用户向量)

Method	Weibo (time)			Weibo (time+user)		
	Precision	Recall	F1	Precision	Recall	F1
SMOP(GRU)	0.851 064	0.941 176	0.893 854	0.904 255	0.876 288	0.890 052
SMOP(LSTM)	0.895 833	0.86	0.877 551	0.893 617	0.933 333	0.913 043

可以看到, 加入了转发用户序列的兴趣向量作为输入后, SMOP(GRU) 模型的性能变化不大, SMOP(LSTM) 模型的性能有明显的提升, 说明了 SMOP 模型的用户兴趣建模方式对预测性能的提升有一定的帮助.

下面我们将实验验证各种算法的性能随指示时间的变化.

图 9 展示了当可观测时间改变时, 算法的 F1 分数随观测时间的变化趋势, 其中, SMOP(G) 和 SMOP(L) 分别为 SMOP(GRU) 和 SMOP(LSTM) 的缩写. Twitter 数据用输入只有转发时间序列, Weibo 数据输入的数据同时包

含了转发时间和用户序列.首先可以看到,SMOP 模型在两个数据上表现出了一致优良的性能,说明 SMOP 对各种数据应用场景具有良好的适应能力;其次,SMOP 的 $F1$ 分数保持在 0.85 上下,对可观测时间的长度变化没有其他比较算法敏感;最后,当可观测时长非常短(0.5h)时,SMOP 模型依然表现了较好的性能,说明 SMOP 模型十分适合于消息爆发预测这种预报时间要求较高的场合.

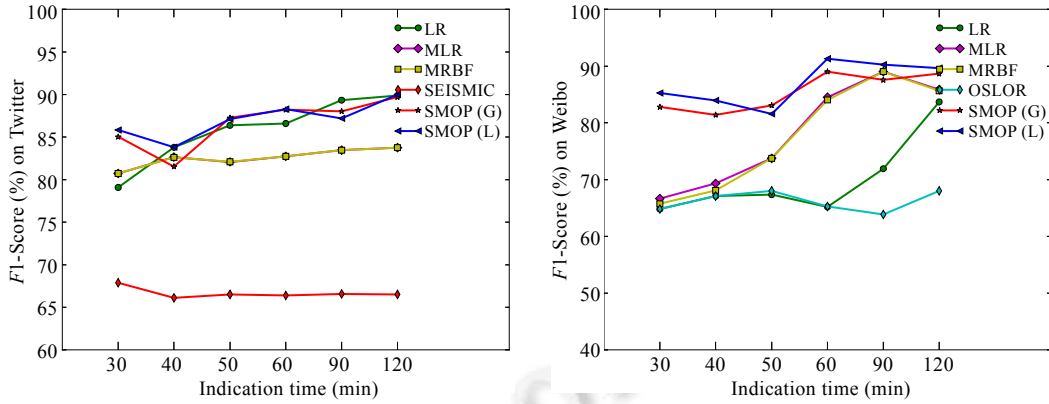


Fig.9 F1-Score with varying indication time

图 9 $F1$ 分数随指示时间变化图示

4.4.2 下一次转发时间预测

为了进一步验证 SMOP 确实学到了消息传播过程的传播速率,我们对 SMOP 模型的底层网络进行消息转发下一次转发时间的预测.从上文的分析中,在每次转发时都可以计算出消息传播的速率函数.

可以用生存模型(survival models)来建模并预测下一次转发时间 T .假设 T 是一个连续的随机变量,表示消息被转发之前的等待时间;累积分布函数(cumulative distribution function,简称 CDF)定义为 $F(t)=Pr\{T<t\}$,表示自从上一次转发时刻 t_i ,消息在时刻 t 之前再次被转发的概率; $f(t)$ 为概率密度函数;生存函数(survival function)定义为 $S(t)=Pr\{T > t\}=1-F(t)$,表示消息在时刻 t 之前没有被转发的概率.

由点过程速率函数定义:

$$\lambda(t) = \lim_{dt \rightarrow 0} \frac{Pr(N(t+dt) - N(t) > 0 | H_t)}{dt} \tag{29}$$

可以推导出点过程的速率函数 $\lambda(t)$ 与生存模型的概率密度函数 $f(t)$ 和生存函数 $S(t)$ 的关系,即

$$\lambda(t) = \lim_{dt \rightarrow 0} \frac{Pr(t < T < t+dt | T > t)}{dt} = \lim_{dt \rightarrow 0} \frac{Pr(t < T < t+dt, T > t)}{dt \times Pr(T > t)} = \frac{f(t)}{S(t)} \tag{30}$$

可得 $\lambda(t) = -d(S(t))/S(t) = -d(\ln(S(t)))/dt$,解微分方程,得到:

$$S(t) = \exp\left(-\int_{t_i}^t \lambda(s) ds\right) \tag{31}$$

因此,

$$f(t) = \lambda(t) \exp\left(-\int_{t_i}^t \lambda(s) ds\right) \tag{32}$$

可见,点过程的 $\lambda(t)$ 和生存模型中的 $f(t)$ 是对应的,这就像 Poisson 分布和指数分布的对应关系.

假设在 t 时刻,根据消息的转发历史,由 SMOP 模型计算得到的点过程的条件速率函数记为 $\lambda(t)$,则下一次消息的转发时间,可以通过生存模型的概率密度函数计算等待时间 T 的期望得到,如下所示.

$$t_{i+1} = \int_{t_i}^{\infty} t \cdot f(t) dt \tag{33}$$

或者

$$t_{i+1} = \int_{t_i}^{\infty} S(t) dt \tag{34}$$

比较的方法为常见的机器学习模型和时间序列分析模型、岭回归模型(ridge regression,简称 RR)、自回归模型(autoregressive,简称 AR)、自回归移动平均模型(autoregressive moving average,简称 ARMA)以及 SEHP^[7]模型,结果使用均方根误差(root mean square error,简称 RMSE)评估,实验结果见表 3.

Table 3 Performance evaluation on next retweeting time prediciton

表 3 下一次转发时间预测性能评估

Method	Twitter	Weibo (time)	Weibo (time+user)
AR(2)	30.47	3.72	-
MA(2)	94.00	31.70	-
ARMA(2,1)	28.80	5.61	-
SEHP	31.91	4.24	-
SMOP(GRU)	28.89	3.71	3.37
SMOP(LSTM)	28.04	3.70	3.37

在表 3 中,AR(2)表示 2 阶自回归模型,MA(2)表示 2 阶移动平均模型,ARMA(2,1)表示 2 阶自回归 1 阶移动平均的组合模型,SEHP 为自激励的 Hawkes 过程.可以看到,SMOP(LSTM)在两个数据集上都取得了最好的性能;SMOP(GRU)在 Weibo 数据集上取得了次优的性能,在 Twitter 数据集上排第 3,比 ARMA(2,1)稍差.AR(2)和 ARMA(2,1)在两个数据集上表现不一致,说明其性能比较依赖于具体的数据集,但总的来说,性能还是可以接受的.SEHP 的性能表现未能超过普通的时间序列分析模型.MA(2)的测试结果最差,说明其可能不适合单独用来进行下一次转发时间的预测任务.另外在微博数据,当 SMOP 模型的输入中加入用户转发序列时,我们发现性能获得了较大的提升,这说明将用户兴趣建模到 SMOP 模型中是有用的.

5 结 论

社交网络的传播动态研究是目前社会计算领域研究的热点之一.本文着重研究了消息爆发预测这个问题,分析了它与流行度预测的差别,接着,利用深度 RNN 网络建模消息的传播过程提出了 SMOP 模型:首先,该模型能够自学习消息传播过程的速率函数,相对于其他手动编码速率特征函数的模型,SMOP 具有良好的数据适应能力;其次,针对神经网络的特点,采用了时间和用户向量化的输入方式,提出了时间和用户的向量化方法,将时间的周期性和用户的兴趣偏好建模到 SMOP 模型中,扩展了点过程模型的输入特征;最后,采用了基于门的循环网络单元,避免了 SMOP 模型训练中的梯度消失问题.在 Twitter 和 Weibo 两个数据集上的实验结果表明, SMOP 在消息爆发预测和下一次消息转发时间预测上都取得了目前较好的成绩.进一步的研究可以尝试采用多目标优化的方式,联合建模优化消息爆发和下一次转发时间预测任务;另外,可采用注意力(attention)机制^[22]对转发用户的重要程度做出精细化的区分,进一步提高模型的性能.

References:

- [1] Myers SA, Leskovec J. The bursty dynamics of the Twitter information network. In: Proc. of the WWW. 2014. 913–924. [doi: 10.1145/2566486.2568043]
- [2] Hong L, Dan O, Davison BD. Predicting popular messages in Twitter. In: Proc. of the WWW. 2011. 57–58. [doi: 10.1145/1963192.1963222]
- [3] Cui P, Jin S, Yu L, Wang F, Zhu W, Yang S. Cascading outbreak prediction in networks: A data-driven approach. In: Proc. of the KDD. 2013. 901–909. [doi: 10.1145/2487575.2487639]
- [4] Kupavskii A, Ostroumova L, Umnov A, Usachev S, Serdyukov P, Gusev G, Kustarev A. Prediction of retweet cascade size over time. In: Proc. of the CIKM. 2012. 2335–2338. [doi: 10.1145/2396761.2398634]
- [5] Pinto H, Almeida JM, Gonçalves MA. Using early view patterns to predict the popularity of youtube videos. In: Proc. of the WSDM. 2013. 365–374. [doi: 10.1145/2433396.2433443]
- [6] Shen H, Wang D, Song C, Barabási A. Modeling and predicting popularity dynamics via reinforced Poisson processes. In: Proc. of the AAAI. 2014. 291–297.

- [7] Bao P, Shen H, Jin X, Cheng X. Modeling and predicting popularity dynamics of microblogs using self-excited hawkes processes. In: Proc. of the WWW. 2015. 9–10.
- [8] Wang D, Song C, Barabási A. Quantifying long-term scientific impact. Science, 2013,342(6154):127–132. [doi: 10.1126/science.1237825]
- [9] Lipton ZC. A critical review of recurrent neural networks for sequence learning. In: Proc. of the Computer Science. 2015. 1–35.
- [10] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997,9(8):1735–1780. [doi: 10.1162/neco.1997.9.8.1735]
- [11] Cho K, Merriënboer BV, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proc. of the EMNLP. 2014. 1724–1734.
- [12] Zhao Q, Erdogdu MA, He HY, Rajaraman A, Leskovec J. SEISMIC: A self-exciting point process model for predicting Tweet popularity. In: Proc. of the KDD. 2015. 1513–1522.
- [13] Szabo G, Huberman BA. Predicting the popularity of online content. Communications of the ACM, 2010,53(8):80–88. [doi: 10.1145/1787234.1787254]
- [14] Bao P, Shen H, Huang J, Cheng X. Popularity prediction in microblogging network: A case study on sina Weibo. In: Proc. of the WWW. 2013. 117–118.
- [15] Gao S, Ma J, Chen Z. Modeling and predicting retweeting dynamics on microblogging platforms. In: Proc. of the WSDM. 2015. 107–116. [doi: 10.1145/2684822.2685303]
- [16] Tatar A, de Amorim M, Fdida S, Antoniadis P. A survey on predicting the popularity of Web content. Journal of Internet Services and Applications, 2014,5(1):8–27. [doi: 10.1186/s13174-014-0008-y]
- [17] Leskovec J, Krause A, Guestrin C, Faloutsos C, VanBriesen J, Glance N. Cost-Effective outbreak detection in networks. In: Proc. of the KDD. 2007. 420–429. [doi: 10.1145/1281192.1281239]
- [18] Gers FA, Schmidhuber J, Cummins F. Learning to forget: Continual prediction with LSTM. Neural Computation, 2000,12(10):2451–2471. [doi: 10.1162/089976600300015015]
- [19] Greff K, Srivastava RK, Koutnik J, Steunebrink BR, Schmidhuber J. LSTM: A search space odyssey. IEEE Trans. on Neural Networks and Learning Systems, 2015,28(10):2222–2232. [doi: 10.1109/TNNLS.2016.2582924]
- [20] Zeiler MD. ADADELTA: An adaptive learning rate method. arXiv preprint, arXiv:1212.5701, 2012.
- [21] Zaremba W, Sutskever I, Vinyals O. Recurrent neural network regularization. In: Proc. of the ICLR. 2013. 1–8.
- [22] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. In: Proc. of the ICLR. 2015. 1–15.



苟程成(1985 -),男,湖北宜昌人,博士,主要研究领域为机器学习,深度学习,网络信息安全.



伍大勇(1977 -),男,博士,高级工程师,主要研究领域为自然语言处理,舆情分析.



秦宇君(1992 -),男,硕士生,主要研究领域为机器学习,自然语言处理.



刘悦(1970 -),女,博士,副研究员,主要研究领域为网络科学与社会计算,互联网搜索,数据挖掘.



田甜(1985 -),女,学士,主要研究领域为网络通信,信息安全.



程学旗(1971 -),男,博士,研究员,博士生导师,CCF 会士,主要研究领域为网络科学与社会计算,网络信息安全.