Collective Opinion Spam Detection : Bridging Review Networks and Metadata

作者：Shebuti Rayana， Leman Akoglu

Stony Brook University Department of Computer Science

KDD 2015

Abstract

提出一种模型 $SPEAGLE$，可以利用元数据、数据关系在一个统一得框架下去对于用户、评论、产品分类。

创新点/贡献/优势：

- 提出SPEAGLE充分利用所有图，行为和评论的内容信息
- SPEAGLE 构建用户-评论-产品的图去完成一个分类问题，利用元数据提取出信息，得到对于分类的先验知识。
- 它是无监督的模式，但是也可以轻松转化为半监督，只需要很少的标注就可以提高分类的准确性。
- 还提出了性能更优的模型SPLITE，使用更少更有效的特征进行处理。

Introduction

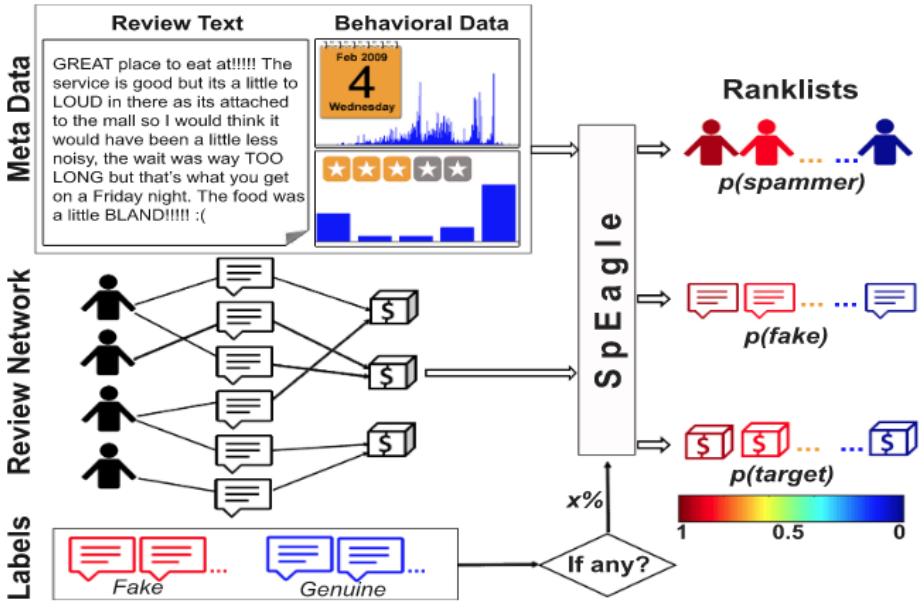


Figure 1: SpEagle collectively utilizes both metadata (review text, timestamp, rating) and the review network (plus available labels, if any) under a *unified* framework to rank all of users, reviews, and products by spamicity.

Model

FRAUDEAGLE Framework

构建一张图，$G = (V, E^{\pm})$ 节点为用户和产品，是一个二分图，利用评论进行连边。

$(ui, pj, s) \in E\pm \ \ s \in \{+, -\}$

$L_U = \{benign, spammer\}$

$L_P = \{good - quality, bad - quality\}$

构建MRF Markov Random Field

$$P(\mathbf{y}) = \frac{1}{Z} \prod_{Y_i \in V} \phi_i(y_i) \prod_{(Y_i, Y_j, s) \in E^{\pm}} \psi_{ij}^{s}(y_i, y_j) \qquad (1)$$

从而计算所有节点分配标签y的一个概率值。

更新的一个转移方式

$\psi^{s=+}$	Product	
User	*good*	*bad*
benign	$1-\epsilon$	ϵ
spammer	2ϵ	$1-2\epsilon$

$\psi^{s=-}$	Product	
User	*good*	*bad*
benign	ϵ	$1-\epsilon$
spammer	$1-2\epsilon$	2ϵ

利用如下信息：

- 相比之下，良性用户通常会对优质产品发表正面评论，对不良质量产品发表负面评论
- 恶意评论者更有可能对劣质产品发表正面评论，或者相反
- 恶意评论者也可以撰写真实的评论来掩饰其欺骗行为
- 对于良性用户而言，对优质产品发表负面评论的可能性很小

但是MRF， 所有可能赋值的枚举都与网络大小成指数关系，因此对于大图来说是很难处理的。

所以采用迭代近似算法LBP （Loopy Belief Propagation）算法

Proposed Method SpEagle

$L_P = \{non - target, target\}$ 由于是好产品还是坏产品不容易判断，并且容易受到影响。

改变转移矩阵为下图

	User ($\psi^{t=\text{`write'}}$)		($\psi^{t=\text{`belong'}}$) Product	
Review	*benign*	*spammer*	*non-target*	*target*
genuine	1	0	$1-\epsilon$	ϵ
fake	0	1	ϵ	$1-\epsilon$

From metadata to features to priors

$x1i, \ldots, xFi$

$$f(x_{li}) = \begin{cases} 1 - P(X_l \leq x_{li}), & if\ high\ is\ suspicious\ (H) \\ P(X_l \leq x_{li}), & otherwise\ (L) \end{cases}$$

$$S_i = 1 - \sqrt{\frac{\sum_{l=1}^{F} f(x_{li})^2}{F}} \qquad (2)$$

S值在0-1之间，表示是否为spam的分数大小

选取的特征值xi

User & Product Features			
Behavior	MNR	H	Max. number of reviews written in a day [18, 20]
	PR	H	Ratio of positive reviews (4-5 star) [20]
	NR	H	Ratio of negative reviews (1-2 star) [20]
	avgRD	H	Avg. rating deviation $avg(\|d_{i*}\|)$ of user (product) i's reviews [5, 15, 20], where $\|d_{ij}\|$ is absolute rating deviation of i's rating from j's average rating: $avg_{e_{ij} \in E_{i*}} \|d_{ij}\|$, for $d_{ij} = r_{ij} - avg_{e \in E_{*j}} r(e)$
	WRD	H	Weighted rating deviation [15], where reviews are weighed by recency: $\frac{\sum_{e_{ij} \in E_{i*}} \|d_{ij}\| w_{ij}}{\sum_{e_{ij} \in E_{i*}} w_{ij}}$, for $w_{ij} = \frac{1}{(t_{ij})^{\alpha}}$ (t_{ij} is rank order of review e_{ij} among reviews of j, $\alpha = 1.5$ is decay rate)
	BST	H	Burstiness [5, 20]—spammers are often short-term members of the site. $x_{BST}(i) = \begin{cases} 0, & \text{if } L(i) - F(i) > \tau \\ 1 - \frac{L(i) - F(i)}{\tau}, & \text{otherwise} \end{cases}$ where $L(i) - F(i)$ is number of days between last and first review of i, $\tau = 28$ days.
	ERD	L	Entropy of rating distribution of user's (product's) reviews [new]
	ETG	L	Entropy of temporal gapsΔ_t's. Given the temporal line-up of a user's (product's) reviews, each Δ_t denotes the temporal gap in days between consecutive pairs [new]
Text	RL	L	Avg. review length in number of words [20]
	ACS	H	Avg. content similarity—pairwise cosine similarity among user's (product's) reviews, where a review is represented as a bag-of-bigrams [5, 15]
	MCS	H	Max. content similarity—maximum cosine similarity among all review pairs [18, 20]

Review Features

Behavior	Rank	L	Rank order among all the reviews of product [9]
	RD	H	Absolute rating deviation from product's average rating [13]
	EXT	H	Extremity of rating [18]: $x_{EXT} = 1$ for ratings $\{4, 5\}$, 0 otherwise (for $\{1, 2, 3\}$)
	DEV	H	Thresholded rating deviation of review e_{ij} [18]: $x_{DEV}(i) = \begin{cases} 1, & if \ \frac{\lvert r_{ij} - avg_{e \in E_{*j}} r(e) \rvert}{4} > \beta_1 \\ 0, & otherwise \end{cases}$ where β_1 is learned by recursive minimal entropy partitioning
	ETF	H	Early time frame [18]—spammers often review early to increase impact. $x_{ETF}(f(e_{ij})) = 1$ if $f(e_{ij}) > \beta_2$, and 0 otherwise, where, $f(e_{ij}) = \begin{cases} 0, & if \ T(i,j) - F(j) > \delta \\ \frac{T(i,j) - F(j)}{\delta}, & otherwise \end{cases}$ where $T(i,j) - F(j)$ is the difference between the time of review e_{ij} and first review j, for $\delta = 7$ months, and β_2 is estimated by recursive minimal entropy partitioning
	ISR	H	Is singleton? If review is user's sole review, then $x_{ISR} = 1$, otherwise 0 [new]
Text	PCW	H	Percentage of ALL-capitals words [9, 13]
	PC	H	Percentage of capital letters [13]
	L	L	Review length in words [13]
	PP1	L	Ratio of 1st person pronouns ('I', 'my', etc.) [13]
	RES	H	Ratio of exclamation sentences containing '!' [13]
	SW	H	Ratio of subjective words (by sentiWordNet) [13]
	OW	L	Ratio of objective words (by sentiWordNet) [13]
	F	H	Frequency of review (approximated using locality sensitive hashing) [new]
	DL_u	L	Description length (information-theoretic) based on unigrams (i.e., words) [new]
	DL_b	L	Description length based on bigrams [new]

Semi-supervised SpEagle

我们只需将与spam的先验知识初始化为 $\{\epsilon, 1 - \epsilon\}$

The algorithm

1 **Input:** User–Review–Product graph $G = (V, E)$, compatibility potentials ψ^t (Table 1), review metada (ratings, timestamps, text), labeled node set L

2 **Output:** Class probabilities for each node $i \in V \backslash L$

首先输入是用户-评论-产品的一张图，对每一条边都赋值，包括有标签的和没标签的。

输出是每一种标签的的可能性

foreach $i \in V$ **do** // *compute/initialize priors*
  if $i \in L$ **then**
    if *i is positive (spam) class* **then** $\phi_i \leftarrow \{\epsilon, 1-\epsilon\}$
    else $\phi_i \leftarrow \{1-\epsilon, \epsilon\}$
  else
    Extract corresponding features in Table 2
    Compute spam score S_i using Eqn. (2)
    $\phi_i \leftarrow \{1-S_i, S_i\}$

对点的概率值，进行赋值，对于有标注的就按照上面的值进行赋值，无标注的就按照公式2，使用提取的先验特征值，进行赋值

foreach $(Y_i^{T_i}, Y_j^{T_j}, t) \in E$ **do** // *initialize all msg.s*
  foreach $y_j \in \mathcal{L}_{T_j}$ **do**
    $m_{i \to j}(y_j) \leftarrow 1$

初始化边的消息值为1

repeat // *iterative message passing*
  foreach $(Y_i^{T_i}, Y_j^{T_j}, t) \in E$ **do**
    foreach $y_j \in \mathcal{L}_{T_j}$ **do**

$$m_{i \to j}(y_j) = \alpha \sum_{y_i \in \mathcal{L}_{T_i}} \left(\phi_i(y_i)\ \psi_{ij}^t(y_i, y_j) \prod_{Y_k \in \mathcal{Y}_{\mathcal{N}_i} \setminus Y_j} m_{k \to i}(y_i) \right)$$

until messages stop changing within a δ threshold

枚举边，通过邻接节点的值更新消息值，直到收敛

foreach $Y_i^{T_i} \in \mathcal{Y}_{V \setminus L}$ **do** // *compute final beliefs*
  foreach $y_i \in \mathcal{L}_{T_i}$ **do**

$$b_i(y_i) = \beta\ \phi_i(y_i) \prod_{Y_j \in \mathcal{Y}_{\mathcal{N}_i}} m_{j \to i}(y_i)$$

对于每个点，计算出每种类型的可能性。

Light-weight SpEagle

原始的 SpEagle计算图中每种类型的每个（未标记）节点的所有特征

确定了评论特征的一小部分根据这些特征计算出的spam分数来初始化未标记评论的优先级，并为（未标记）用户和产品使用无偏先验{0.5,0.5}。 这大大减少了评论的特征提取开销，并完全避免了用户和产品使用它，从而仅在分类效果略有妥协的情况下实现了加速

Experiments

数据集

Table 3: Review datasets used in this work.

Dataset	#Reviews (filtered %)	#Users (spammer %)	#Products (rest.&hotel)
YelpChi	67,395 (13.23%)	38,063 (20.33%)	201
YelpNYC	359,052 (10.27%)	160,225 (17.79%)	923
YelpZip	608,598 (13.22%)	260,277 (23.91%)	5,044

Yelp数据集包含由Yelp过滤和推荐的酒店和餐厅评论。

Table 4: AP and AUC performance of compared methods on all three datasets.

	User Ranking						Review Ranking					
	AP			AUC			AP			AUC		
	Y'Chi	Y'NYC	Y'Zip	Y'Chi	Y'NYC	Y'Zip	Y'Chi	Y'NYC	Y'Zip	Y'Chi	Y'NYC	Y'Zip
RANDOM	0.2024	0.1782	0.2392	0.5000	0.5000	0.5000	0.1327	0.1028	0.1321	0.5000	0.5000	0.5000
FRAUDEAGLE	0.2537	0.2233	0.3091	0.6124	0.6062	0.6175	0.1067	0.1122	0.1524	0.3735	0.5063	0.5326
WANG ET AL.	0.2659	0.2381	0.3306	0.6167	0.6207	0.6554	0.1518	0.1255	0.1803	0.5062	0.5415	0.5982
PRIOR	0.2157	0.1826	0.2550	0.5294	0.5081	0.5269	0.2241	0.1789	0.2352	0.6707	0.6705	0.6838
SPEAGLE	**0.3393**	**0.2680**	**0.3616**	**0.6905**	**0.6575**	**0.6710**	**0.3236**	**0.2460**	**0.3319**	**0.7887**	**0.7695**	**0.7942**
SPEAGLE$^+$(1%)	0.3967	0.3480	0.4245	0.7078	0.6828	0.6907	0.3352	0.2757	0.3545	0.7951	0.7829	0.8040
SPLITE$^+$ (1%)	0.3777	0.3331	0.4218	0.6744	0.6542	0.6784	0.3124	0.2550	0.3448	0.7693	0.7631	0.7923

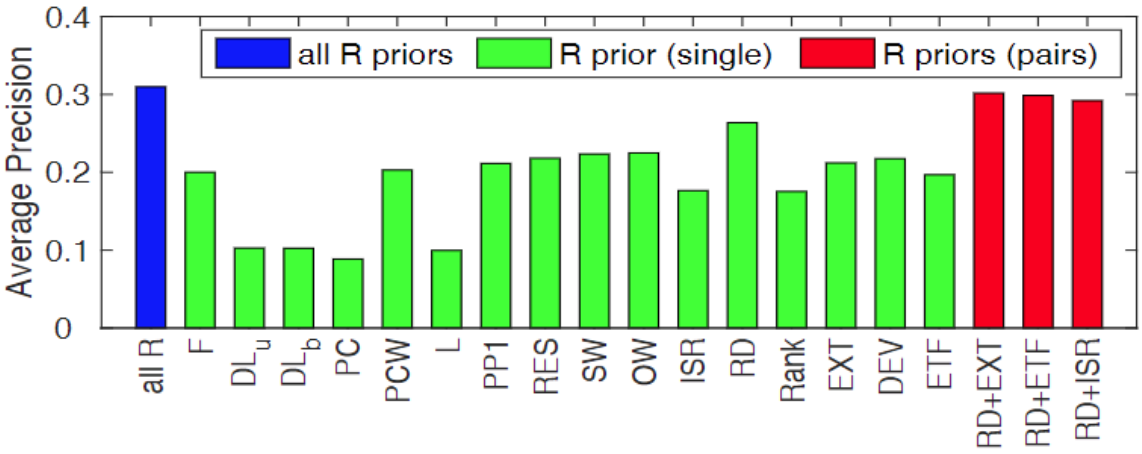
基本上达到最佳效果

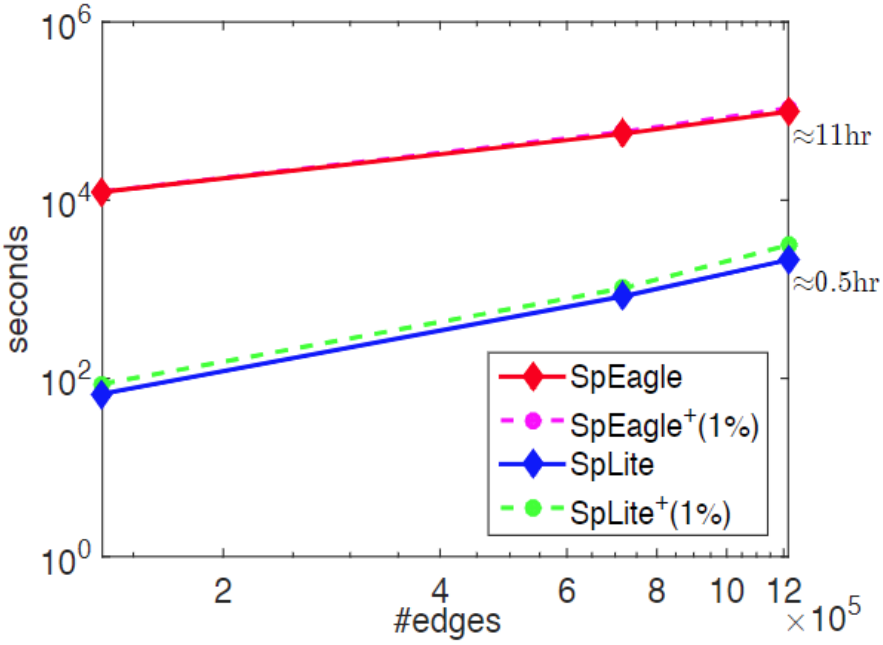
	YelpChi				YelpNYC				YelpZip			
k	0%	1%	5%	10%	0%	0.5%	1%	2%	0%	0.25%	0.5%	1%
100	0.7400	0.9300	0.9650	0.9950	0.4400	0.9650	0.9630	0.9930	0.4300	0.8740	0.8540	0.9090
200	0.5900	0.8195	0.9565	0.9600	0.4600	0.9595	0.9625	0.9790	0.5150	0.8850	0.8935	0.9130
300	0.5333	0.6910	0.9477	0.9500	0.4433	0.9557	0.9553	0.9713	0.5133	0.8303	0.9037	0.9173
400	0.4975	0.6162	0.9245	0.9408	0.4350	0.8935	0.9587	0.9710	0.5250	0.7823	0.8972	0.9225
500	0.5020	0.5736	0.8772	0.9344	0.4100	0.8076	0.9586	0.9664	0.5260	0.7574	0.8750	0.9212
600	0.4900	0.5617	0.8008	0.9110	0.3983	0.7602	0.9603	0.9633	0.5150	0.7320	0.8500	0.9218
700	0.4600	0.5407	0.7451	0.8671	0.3943	0.7079	0.9521	0.9623	0.4971	0.7090	0.8307	0.9226
800	0.4587	0.5125	0.7015	0.8078	0.3900	0.6685	0.9067	0.9616	0.4900	0.6946	0.8138	0.9178
900	0.4544	0.5018	0.6739	0.7570	0.3844	0.6307	0.8586	0.9610	0.4833	0.6711	0.7938	0.9106
1000	0.4510	0.4944	0.6471	0.7141	0.3820	0.5982	0.8225	0.9597	0.4880	0.6453	0.7744	0.9004

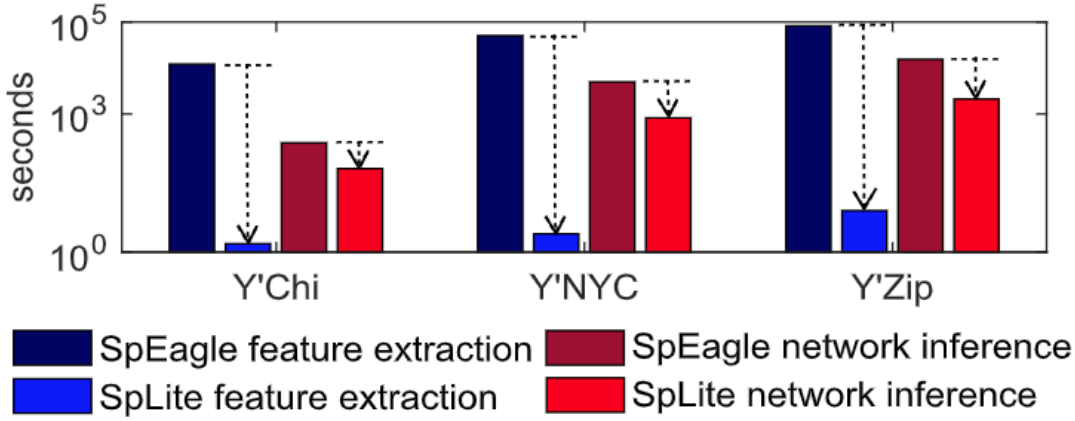
使用半监督，明显提高分类效果

	User Ranking						Review Ranking					
	AP			AUC			AP			AUC		
	Y'Chi	Y'NYC	Y'Zip	Y'Chi	Y'NYC	Y'Zip	Y'Chi	Y'NYC	Y'Zip	Y'Chi	Y'NYC	Y'Zip
RANDOM	0.2024	0.1782	0.2392	0.5000	0.5000	0.5000	0.1327	0.1028	0.1321	0.5000	0.5000	0.5000
SPEAGLE (U)	0.3197	0.2624	0.2808	0.6767	0.6483	0.6183	0.3043	0.2400	0.1427	0.7783	0.7629	0.5940
SPEAGLE (P)	0.1550	0.1357	0.1814	0.3905	0.3930	0.3801	0.0755	0.0640	0.0806	0.1643	0.2536	0.2277
SPEAGLE (R)	0.3226	0.2575	0.3449	0.6771	0.6477	0.6562	0.3098	0.2378	0.3180	0.7820	0.7656	0.7884
SPEAGLE (UR)	**0.3398**	**0.2680**	0.3615	**0.6905**	**0.6575**	0.6709	**0.3241**	**0.2460**	**0.3320**	**0.7887**	**0.7695**	**0.7942**
SPEAGLE (URP)	0.3393	**0.2680**	**0.3616**	**0.6905**	**0.6575**	**0.6710**	0.3236	**0.2460**	0.3319	**0.7887**	**0.7695**	**0.7942**

使用评论或者用户可以更好的分类，只使用产品，达到的效果还没有随机好







Conclusion

在这项工作中，提出了一个名为SpEagle的新整体框架，该框架可以选择性地利用关系数据（用户-评论-产品图）和元数据（行为和文本数据）来检测可疑的用户和评论以及spam所针对的产品。

主要的工作是

- SpEagle实施基于二分网络的分类任务，该任务接受从元数据估计的节点的类分布的先验知识。
- SpEagle 以无人监督的方式工作，但可以轻松利用标签（如果有）。 因此，我们引入SpEagle +的半监督版本，可显着提高性能。
- 我们进一步设计了名为SpLite的SpEagle的light version，该版本使用了很少的审核功能作为先验信息，从而显着提高了速度。

在三个真实的数据集上评估了我们的方法 带有标记的评论（已过滤与推荐），从Yelp.com收集。因此，我们提供了迄今为止针对恶意评论检测的最大规模的定量评估结果。

我们的结果表明，SpEagle优于几种方法和最新技术。