Crowd signals for Fake news Detection

Vansh Kapoor
Guide: Prof. Nikhil Karamchandani

Department of Electrical Engineering
IIT Bombay

Outline

Motivation

- Project goal is to minimize the spread of misinformation in social networks by stopping the spread of fake news in this network.
- The vast volume spread of news make traditional human-based verification impractical
- Model capable of FND among vast dynamically generated news by using true labels from a smaller sample set.

Previous Work

- *Curb* leverages flagging activity of users to detect fake news Treats all flags as equally reliable
- Recent work involves modeling network structures like ICM and NLP techniques
- Challenges: Limited availability of corpora and the substantial variability in the sources of fake news.

Novelty

- Unlike any other approach *Detective* uses a semi-supervised crowd sourcing technique that allows users' responses to be verified by external experts.
- Sebastian et al. proposed an Online algorithm *Detective* that uses Bayesian inference to detect fake news and concurrently learns about the user's flagging accuracy.
- Unlike *Curb* which treats all the flags as equally reliable *Detective* learns about the flagging accuracy of every user in an Online setting.

Overview of the Approach

1. Formulation of the Objective Function
2. Online Learning of Flagging Accuracy
3. *Detective*: Proposed Model

Formulation of the Objective Function

Setting Up Notation

- Let $Y^*(x)$ be the random variable denoting the realization of label $y^*(x) \in \{f, \bar{f}\}$ of news x, where f and $\bar{f}$ are labels for "fake" and "not fake" respectively.
- Let $\pi^t(a)$ be the set of users who have seen the news $a \in A^t$, where A^t denotes the set of active news generated by the end of epoch $t \geq 1$.
- Furthermore, let $\psi^t(a)$ be the set of users who have flagged news $a \in A^t$ as fake.
- Also let $\theta_{u,\bar{f}} := P(Y_u(x) = \bar{f} | Y^*(x) = \bar{f})$ and $\theta_{u,f} = P(Y_u(x) = f | Y^*(x) = f)$

Formulation of the Objective Function

Overview of a Generalized Algorithm

- At the beginning of every epoch t we set $A^t = A^{t-1} \cup X^t$
- At the end of every epoch t, we choose $S^t \subseteq A^t$ and obtain the true label for every element.
- Subsequently we perform the update $A^t = A^t / S^t$
- Observe that $|\pi^\infty(a)|$ gives the number of users affected by news a is it is allowed to spread through the network
- We define the objective function for an algorithm *Algo* with a horizon of T as

$$Util(T, Algo) = \sum_{t=1}^{T} \mathbb{E}\left[\sum_{s \in S_t} \mathbf{1}_{\{y^*(s)=f\}} \text{val}^t(s)\right]$$

where $\text{val}^t = |\pi^\infty(a)| - |\pi^t(a)|$

Online Learning of Flagging Accuracy

For a prior distribution of the user's parameters, denoted as $(\Theta_f, \Theta_{\bar{f}})$, we define $\mathcal{D}_u^t$ as the data history given by

$$\mathcal{D}_u^t = \begin{pmatrix} d^t_{u,\bar{f}|\bar{f}} & d^t_{u,\bar{f}|f} \\ d^t_{u,f|\bar{f}} & d^t_{u,f|f} \end{pmatrix}$$

Here, $d^t_{u,f|\bar{f}}$ is the number of times user u flagged a verified correct news as fake.

$$P(\theta_{u,\bar{f}}|\Theta_{\bar{f}}, \mathcal{D}_u^t) = (\theta_{u,\bar{f}})^{d^t_{u,\bar{f}|\bar{f}}} \cdot (1-\theta_{u,\bar{f}})^{d^t_{u,f|\bar{f}}} \cdot P(\theta_{u,\bar{f}}|\Theta_{\bar{f}})$$

It is easy to check that the solution to these is given by the $Beta_{d^t_{u,\bar{f}}, d^t_{u,f|\bar{f}}}(\theta_{u,\bar{f}})$ distribution.

Note: Selecting user parameters using MAP is not recommended since finite exploration can lead to non-optimal solutions even for $T \to \infty$

Detective: Proposed Model

Algorithm Detective

1: Sample $\theta_{u,\bar{f}} \sim P(\theta_{u,\bar{f}}|\Theta_{\bar{f}}, \mathcal{D}_u^t)$ for all $u \in U$ wrt to the Beta distribution mentioned above
2: Sample $\theta_{u,f} \sim P(\theta_{u,f}|\Theta_f, \mathcal{D}_u^t)$ for all $u \in U$ wrt to the Beta distribution mentioned above
3: Use Bayesian inference wrt the above sampled user parameters to choose $S^t \subseteq A^t$ to maximize the objective function *Util*
4: **Return** S_t

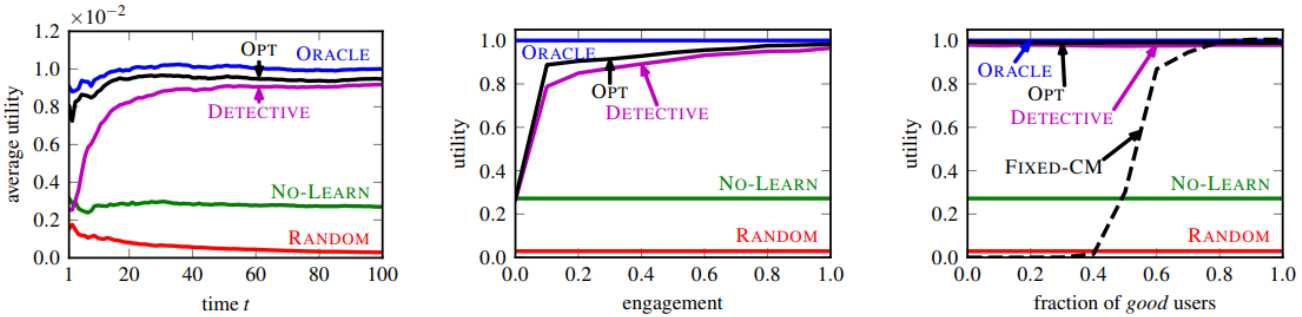
Regret Analysis

$Regret(T, \text{Algo}) = \text{Util}(T, \text{Opt}) - \text{Util}(T, \text{Algo})$

Proposition 1. Any algorithm, Algo, using deterministic point estimates for the users' parameters suffers linear regret, i.e., $Regret(T, \text{Algo}) = \Theta(T)$.

Theorem 1. The expected regret of our algorithm, Detective, is $E[\text{Regret}(T, \text{Detective})] = \mathcal{O}\left(C\sqrt{pM'T\log(CM'T)}\right)$, where $M' = \binom{M}{k}$ and C is a problem-dependent parameter. C quantifies the total number of realizations of how M news can spread to U users and how these users label the news.

Main Results



(a) Learning about users (b) Users' engagement in flagging. (c) Robustness against spammers

Figure 2: Experimental results. ***(a)*** **Learning about users: DETECTIVE achieves average utility competitive compared to that of ORACLE (which knows the true news labels). The average utility of DETECTIVE converges to that of OPT as DETECTIVE progressively learns the users' parameters.** ***(b)*** **Users' engagement in flagging: even with low engagement DETECTIVE can effectively leverage crowd signals to detect fake news.** ***(c)*** **Robustness against spammers: DETECTIVE is effective even if the majority of users is adversarial, highlighting the importance of learning about users' flagging accuracy for robustly leveraging crowd signals.**

Figure: Results from Sebastian et al.

- Experiments demonstrate that Detective is competitive even with an algorithm that know the true user parameters.
- Detective algorithm can effectively leverage crowd signals to detect fake news even in situations of low user engagement.
- Detective is effective even if the majority of users is adversarial which underscores the importance of learning about the user parameters.

Influence Maximization

- We'll now explore a setting of influence maximization that is entirely opposite but remarkably analogous to the previous setting.
- This algorithm learns about a follower's preferences by assigning different weights to the feedback from each follower.
- Similar to *Detective*, by adopting *posterior sampling* and point based strategies the algorithm achieves a logarithmic regret and linear regret respectively.
- Interestingly, by utilizing the feedback that her followers give to other users, she can achieve a $O(1)$ by posterior sampling, and a $o(T)$ regret with using point estimates.

Previous Work

- The traditional Bandit algorithms, such as *Detective* utilize only the feedback that a user receives from her followers.
- The novelty in the approach lies in learning about the followers, utilizing their feedback provided to **other influencers** as well.
- Unlike **Smart broadcasting**, the objective of this setting is what message to share rather than when a message has to be shared

Problem Formulation

- Let $\mathcal{N}(u)$ be the set of followers for an influencer u.
- The users shares topic $c_t \in \mathcal{C}$, $|C| = K$ at every time-step $t \in \{1, \ldots, T\}$ and each follower decides whether to give or not to give feedback.

$$\mathcal{H}_v(t) = \{(c_i, I(c_i)) | i \in \{1, \ldots, T\}\}$$

$I(c_i) = \mathcal{I}\{v \text{ gave feedback to } c_i\}$ and $p_v(I(c) = 1|c) = q_{cv}$

- At every time step $c_t \sim p(c|\mathcal{H}(t)$, where $\mathcal{H}_t = (\mathcal{H}_v(t))_{v \in N(u)}$ and each follower gives her feedback according to $\mathbf{q}_c = (q_{cv})_{v \in \mathcal{N}(u)}$ and $\mathbf{Q} = (\mathbf{q}_c)_{c \in \mathcal{C}}$

Utility Maximization

$$\text{UTIL}(T) = \mathbb{E}\left[\sum_{t\in[T]}\left(\sum_{v\in\mathcal{N}(u)} a_v l_v(c_t) + a_u\right)\right]$$

a_v: Importance u gives to the feedback of follower v
a_u: Importance u gives to her own preferences
x_{c_t}: User u's preference for topic c_t
Note: $0 < x_{c_t} \leq 1$ and $\sum_{v\in\mathcal{N}(u)} a_v + a_u = 1$

$$\begin{aligned}\text{UTIL}(T) &= \sum_{t\in[T]}\left(\sum_{v\in\mathcal{N}(u)} \mathbb{E}_{c_t\sim p^*}\left[\mathbb{E}_{l_v(c_t)|c_t}[a_v l_v(c_t) + a_u]\right]\right)\\ &= \sum_{t\in[T]}\sum_{c\in\mathcal{C}} p^*(c|\mathcal{H}_t)(\mathbf{a}\mathbf{q}_c + a_u x_c)\end{aligned}$$

Utility Maximization

Objective:

$$\max_{p(c|\mathcal{H}(t))} \sum_{t\in[T]} \sum_{c\in\mathcal{C}} p(c \mid \mathcal{H}_t)(\mathbf{a}\mathbf{q}_c + a_u x_c)$$

$$\text{subject to} \quad 0 \leq p(c \mid \mathcal{H}(t)) \leq 1 \quad c \in \mathcal{C}$$

$$\sum_{c\in\mathcal{C}} p(c \mid \mathcal{H}(t)) = 1$$

If user preferences known, we simply select

$$p(c|\mathcal{H}(t)) = \mathbb{I}\left[c = \operatorname*{argmax}_{c'}(\mathbf{a}\mathbf{q}_{\mathbf{c}'} + a_u x_{c'})\right]$$

Unknown user preferences **Q**

$\forall c \in \mathcal{C}$ and $v \in \mathcal{N}(u)$ assume a $p_{cv}(0) \sim Beta(\alpha, \beta)$

$$p(q_{cv}|\mathcal{H}_v(t)) = Beta(\alpha + n_{cv}(t), \beta + \bar{n}_{cv}(t)$$

where $n_{cv}(t)$ are the number of likes given to user u by follower v for a topic in c until time t, i.e., no. of times $(c, 1)$ appears in $\mathcal{H}_v(t)$.

Similarly, $\bar{n}_{cv}(t)$ is the number of times $(c, 0)$ appears in $\mathcal{H}_v(t)$

Unknown user preferences **Q**

1. For *point estimates*: we take the value of the random variable corresponding to the highest value in the pdf

$$\hat{q}_{cv}(t) = argmax p(q_{cv}(t)|\mathcal{H}_v(t)) = \frac{\alpha + n_{cv} - 1}{\alpha + \beta + n_{cv}(t) + \bar{n}_{cv}(t) - 2}$$

2. via sampling from posterior,

$$\hat{q}_{cv}(t) \sim p(q_{cv}(t)|\mathcal{H}_v(t))$$

Using these estimates we choose,

$$p(c|\mathcal{H}(t)) = \mathbb{I}\left[c = \underset{c'}{\text{argmax}}(\mathbf{a}\hat{\mathbf{q}}_{\mathbf{c}'} + a_u x_{c'})\right]$$

Algorithm For Utility Maximization

Algorithm UTIL$_{max}$

Input: Prior α and β
Output: Topic $c_{t\,t\in[T]}$
for $v \in \mathcal{N}_u$:
  $\mathcal{H}_v \leftarrow \phi$
for $t \in \{1, 2, \ldots T\}$:
  $c_t = argmax_{c'}(\sum_{v\in\mathcal{N}(u)} a_v \hat{q}_{c'v}(t) + a_u x_{c'}$
  $Share(c_t)$
  for $v \in \mathcal{N}(u)$:
    $\mathcal{H}_v(t+1) \leftarrow \mathcal{H}_v(t) \bigcup GATHERFEEDBACK(v)$
    for $c \in \mathcal{C}$:
      $\hat{q}_{cv}(t) = ESTIMATE(\alpha, \beta, c, \mathcal{H}_v(t)$

Regret Analysis

$$R(T) = \text{UTIL}^*(T) - \text{UTIL}(T)$$

$\text{UTIL}^*(T)$ is the regret achieved by UTIL_{max} when each users preferences q_{cv} are known and it utilizes q_{cv} instead of $\hat{q}_{cv}$ for choosing c_t

Note:

(i) $\mathcal{H}_v(t)$ could also contain the feedback that followers give to other users on a topic.

(ii) $\text{UTIL}(T)$ uses $\hat{q}_{cv}(t)$ which depends on the method of estimation, i.e., either point estimates or posterior sampling.

Regret Analysis for Point Estimates

Point Estimates

Theorem 1: Assume user u uses point estimates $\hat{q}_{cv}(t)$ and she can only access the feedback she receives from her followers. Then, UTIL$_{max}$ suffers linear regret $\Theta(T)$.

Theorem 2: Assume user u uses point estimates for the followers' preferences $\hat{q}_{cv}(t)$ and she can access both the feedback her followers give to her as well as to others. Furthermore, the amount of feedback each of her followers v give to others per topic c follows a Poisson distribution with rate $\mu_{cv} > 0$, and let $d = \alpha + \beta$. Then, for $2 \leq d < 3$, Algorithm 1 achieves regret

$$\mathcal{O}\left(\sum_{c \in C, v \in \mathcal{N}(u)} \frac{\sqrt{T-1}\sqrt{1-e^{-\mu_{cv}(T-1)}}}{2\sqrt{(d-2)\mu_{cv}}}\right)$$

and, for $d \geq 3$, it achieves:

$$\mathcal{O}\left(\sum_{c \in C, v \in \mathcal{N}(u)} \frac{T-1}{2\sqrt{(T-1)\mu_{cv} + d - 3 + \sqrt{d-3}}}\right)$$

.

Regret Analysis for Posterior Sampling

Posterior Sampling

Theorem 3. Assume user u uses posterior samples to estimate the followers' preferences $\hat{q}_{cv}(t)$, she can access both the feedback her followers give to her as well as to others, and the amount of feedback each of her followers v give to others per topic c follows a Poisson distribution with rate $\mu_{cv} > 0$. Then, Algorithm 1 has regret

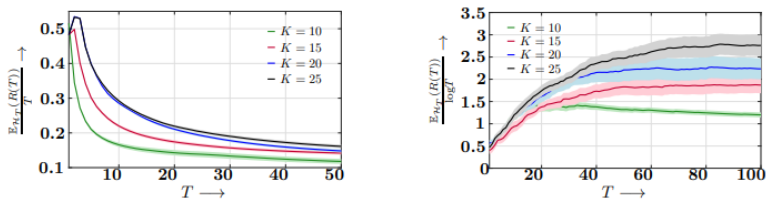
$$\mathcal{O}\left(\log\left(1 + \sum_{v \in \mathcal{N}(u)} \frac{1 - \exp(-\mu_{cv}\theta_m T)}{\mu_{cv}\theta_m}\right)\right)$$

, where θ_m depends on the parameters Q in a non-trivial way.

Theorem 4. Assume user u uses posterior samples to estimate the followers' preferences $\hat{q}_{cv}$ and she can only access the feedback she receives from her followers. Then, Algorithm 1 has regret $\mathcal{O}(log T)$.

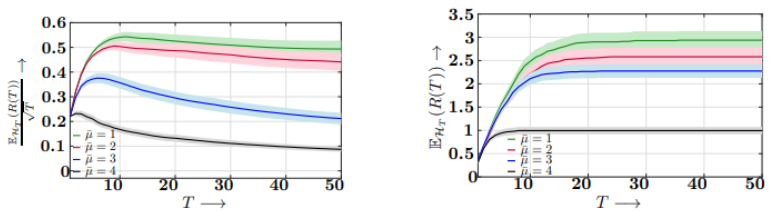
Simulation Results on Regret Analysis

For the experimental setup we consider $|\mathcal{N}(u)| = 10$ followers with $a_v \sim D(\gamma)$ and $a_u \sim D(\gamma)$ with $\gamma = 0.8$, the preference of her followers $x_c \sim Beta(0.4, 0.6)$. We vary μ_{cv} and the number of topics $|\mathcal{C}| = K$ in our experiments.
For the case when u additionally utilizes the feedback her followers give to others we set $K = 10$, sample $\mu_{cv} \sim Unif[0, 2\bar{\mu}]$.



(a) $\hat{q}_{cv}(t)$ are point estimates (b) $\hat{q}_{cv}(t)$ are posterior samples

Figure 1: Regret analysis when user u only leverages the feedback she receives, *i.e.*, $\mu_{cv} = 0$. Panel (a) shows that, if she uses point estimates for her followers' preferences, Algorithm 1 suffers linear regret. Panel (b) shows that, if she uses posterior sampling, it achieves logarithmic regret. In both panels, as the number of topics K increases, the regret increases.



(a) $\hat{q}_{cv}(t)$ are point estimates (b) $\hat{q}_{cv}(t)$ are posterior samples

Figure 2: Regret analysis when user u levegares the feedback she receives as well as the feedback her followers give to others, *i.e.*, $\mu_{cv} > 0$. Panel (a) shows that, if she point estimates for her followers' preferences, Algorithm 1 achieves sublinear regret $O(\sqrt{T})$. Panel (b) shows that, if she uses posterior sampling, it achieves constant regret $O(1)$. In both panels, as the average rate of feedback to others μ_{cv} increases, the regret decreases.

Figure: Regret Analysis Results from Abir De et al.

Utility Estimation Framework

- Our goal in this section is to to determine whether the user utilizes the feedback from each of her followers to decide what to post next.
- Here we find weights $(a_v)_{v \in \mathcal{N}(u)}$ and a_u and parameters $(x_c)_{c \in \mathcal{C}}$ that best fit the observed data

Point Estimates

$$\begin{aligned} \underset{\mathbf{a}, a_u}{\text{minimize}} \quad & \sum_{t \in [T]} \left(\max_{c' \in C} \left(\mathbf{a} \hat{q}_{c'}(t) + a_u x_{c'} \right) - \left(\mathbf{a} \hat{q}^t_{ct}(t) + a_u x_{c_t} \right) \right), \\ \text{subject to} \quad & a_u \geq 0, a_v \geq 0 \quad \forall v \in \mathcal{N}(u), \\ & \sum_{v \in \mathcal{N}(u)} a_v + a_u = 1, \\ & 0 \leq z_c \leq a_u, \quad \forall c. \end{aligned}$$

Utility Estimation Framework

Posterior Sampling

$$\begin{aligned}
\underset{\mathbf{a}, a_u}{\text{minimize}} \quad & \sum_{t\in[T]} \mathbb{E}_{\hat{q}(t), t\in[T]} \left[\left(\max_{c'\in C} \left(\mathbf{a}\hat{q}_{c'}(t) + a_u x_{c'} \right) - \left(\mathbf{a}\hat{q}^t_{ct}(t) + a_u x_{c_t} \right) \right] , \\
\text{subject to} \quad & a_u \geq 0, a_v \geq 0 \quad \forall v \in \mathcal{N}(u), \\
& \sum_{v\in\mathcal{N}(u)} a_v + a_u = 1, \\
& 0 \leq z_c \leq a_u, \quad \forall c.
\end{aligned}$$

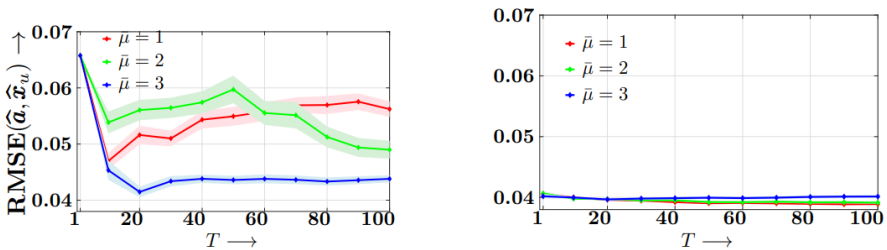
In practice we approximate the objective function by empirical average by

$$\sum_{t\in[T]} \sum_{s=1}^{S} \left[\max_{c_0\in C} \left(\mathbf{a}\hat{q}^{(s)}_{c'}(t) + a_u x_{c'} \right) - \left(\mathbf{a}\hat{q}^{(s)}_{ct}(t) + a_u x_{c_t} \right) \right] ,$$

where S is the number of samples and $\hat{q}^{(s)}cv(t) \sim p(q_{cv}(t)|\mathcal{H}_v(t)) \ \forall c \in \mathcal{C}$

Simulation Results for Utility Estimation Framework

For utility estimation framework, we first generate $H(T)$ by simulating data from our model with $K = 10$ and sample $_{cv} \sim Unif[0, 2]$. The model is trained using the generated $H(T)$, for different T and $\bar{\mu}$ values using our two estimation methods



(a) $\hat{q}_{cv}$ are point estimates (b) $\hat{q}_{cv}$ are posterior samples

Figure 3: Performance of our utility estimation method based on linear loss minimization in terms of $RMSE = \sqrt{\mathbb{E}(||\boldsymbol{a} - \hat{\boldsymbol{a}}||^2) + \mathbb{E}(||a_u - \hat{a}_u||^2) + \mathbb{E}(||\boldsymbol{x}_u - \widehat{\boldsymbol{x}}_u||^2)}$ as T increase. The performance is significantly better whenever the user leverages posterior samples for her followers' preferences.

Figure: Utility Estimation Framework Results from Abir De et al.

Analogy

- Algorithm *Detective* using each users accuracy, aims to selects fake news that is most likely to cause the "most damage".
- Here at every time step our objective is to select the topic that is most likely to be given a feedback (positive).
- The underlying principle of learning each user's reliability (early rumor detection) or liking for a topic (influence maximization) individually remains the same.

Future Work

- The problem on influence maximization has assumed that the followers' preferences are not influenced by the users' posting behavior
- We are yet to explore a scenario in which both users and the followers influence each other
- Also the Utility function here is a simple linear utility function, which neglects the correlation between different followers.
- A natural extension would be to consider more complex utility functions with higher predictive power