Improving Monitoring of Participatory Civil Issue Requests through Optimal Online Classification

Daphney–Stavroula Zois*, Christopher Yong†, Charalampos Chelmis†, Angeliki Kapodistria* and Wonhyung Lee‡
*Department of Electrical and Computer Engineering, †Department of Computer Science, ‡School of Social Welfare
University at Albany, SUNY, Albany, NY, USA
Emails: {dzois, cyong, cchelmis, akapodistria, whlee}@albany.edu

***Abstract*—Participatory civil issue monitoring has emerged as an easy way for concerned citizens to report problems to their local government. For reported issues to be timely processed and addressed however, accurate, online and real–time processing methods to infer issue types are necessary. To address this challenge, we propose a computational, near–real–time civil issue reports processing method to estimate the actual issue from ambiguous and/or complementary information accurately and efficiently. We demonstrate the effectiveness of the proposed approach using a real–world dataset from SeeClickFix. We show that our approach is both highly accurate and scalable.**

I. Introduction

The wide spread of Internet–enabled, location–aware, smart phone devices over the past 15 years has enabled all kinds of users, regardless of their technical background and competency, to gather and share data. This has led to new "Government 2.0" applications [1], through which citizens can actively participate to e.g., measure air quality [2], map fuel consumption on city streets [3], predict bus arrival times [4], and even hunt for grocery bargains [5]. On the other hand, platforms for crowdsourced, mobile participatory civic issue reporting such as FixMyStreet [6] in the UK and SeeClickFix [7] in the US, have emerged lately to assist concerned citizens to report problems to government agencies regarding their local environment through easy–to–use technology.

The possibility to be heard on issues and the ability to actively shape and connect to the urban spaces they live in provides citizens with a strong intrinsic motivation to enhance their living environment, resulting in a high degree of participation in civil issue monitoring [7]–[9]. Administrative bodies and policy makers can utilize such participatory sensing data to gather information on civil issues in urban spaces, however, they can count on the continuous engagement of concerned citizens only if issues reported are timely processed and addressed. Even though the automatic classification of issues [10], their significance [11] and duplicate issues identification [12] have been previously explored, the scalability and timeliness of such methods have largely been ignored. Moreover, the concreteness of reported issues depends on the reporter; the actual status and demand may not be described clearly or either one may be misdescribed in the report, leaving officials scrambling about what the actual problem may be. Finally, currently, each request must be manually evaluated and acknowledged by a city official before being routed to the appropriate agency or maintenance crew that is sent out to fix the issue. Needless to say, this approach does not scale.

This research has been funded by NSF ECCS-1737443.

To address the aforementioned challenges, we propose a computational, near–real–time civil issue reports processing approach to estimate the actual issue from ambiguous and/or complementary information such as textual descriptions and photographs and assign reported issues to the appropriate authorities accurately and efficiently. We formulate this problem as a sequential hypothesis testing problem, in which features extracted from issue reports are examined to infer the type of issue as quickly as possible while ensuring that the risk of missclassification is low. We show that the optimal strategy in this decision problem is an optimal stopping rule: features are sequentially reviewed starting from the most informative, and at each step, the framework decides when to stop. Once stopped, it can classify an issue based on features examined thus far, and "safely" ignore the remaining features. Unlike state–of–the–art classifiers that rely on a fixed set of features for classification once trained, the optimal number of features used by our approach to categorize a reported issue is a function of the cost corresponding to the time and effort spent evaluating each feature, and the classification quality. Given the limited memory and time requirements of real–world systems, our approach provides a viable, realistic solution to participatory civil issue monitoring by efficiently utilizing computational resources rather than invariably applying a "brute force" classification using all features for all issues.

II. SeeClickFix Platform & Data Collection

SeeClickFix, a community advocacy tool designed to bridge the communication gap between residents and their local governments about non–emergency issues (e.g., parking violation or need for snow removal), allows citizens to collectively improve their communities by simply "*Taking a photo of a pothole or other problem, geo–locate it and hit submit. SeeClickFix publicly documents the issue and notifies local governments and others who resolve the problem*" [7]. Other users can view and support issues in the form of "Thanks" votes (similar to the "like" functionality in online social media). Authorities (i.e., a verified account associated with a city official) acknowledge the issue (and, if needed, direct it to the appropriate agency or maintenance crew), which is subsequently resolved, at which time it is marked as "closed".

TABLE I: Information associated with an issue in SeeClickFix.

Data field	Description
Issue ID	A unique 6-to-7 digit ID for each posted issue
Title	Title of the issue
Status	Signifies attention paid by authorities; one of "Open", "Acknowledged", or "Closed"
Address	Location of the issue
Image	A user–provided photo (limit of one per issue)
Reporter ID	A unique 4–to–6 digit ID for registered users
Reporter Name	Screen name of registered users
Votes Count	Number of up–votes the issue received from users
Thanks Count	Number of "Thanks" the issue received from users
Category	The type the issue belongs to
Reported Time	Date and time in UTC $\pm$ 0
Reported Via	Medium used to report an issue
Tags	User–defined keywords for the purpose of simplifying the discovery of "similar" issues by other users
Description	Short comment provided by the user reporting the issue
Q&A	Answers to predefined questions
Total Comments	Total number of comments
Comment List	Comments associated to an issue; each comment has (i) a unique 6–to–8 digit ID, (ii) ID of the user who commented, (iii) status, (iv) image (used to provide additional photos), (v) comment text, (vi) time

In this work, we collected a total of $2,195$ SeeClickFix issues for the metropolitan area surrounding Albany, the capital of the U.S. state of New York, spanning a time period between Jan 5, 2010 and Feb 10, 2018. Albany is the 4rth largest metropolitan region in the state and the 45th largest in the US. Even though issues are publicly available through SeeClickFix[1], we provide a clean version of our dataset on our website[2] to improve the reproducibility of our results, and to promote sustainable and comparable research in the future. Each issue comprises information summarized in Table I. In total, there are 34 categories broadly divided into genres, including but not limited to, parking enforcement, repairs (e.g., potholes or overgrown trees), trash, parks and recreational areas, noise, and housing. Despite the fact that a large portion of the reports (68.6%) have been manually categorized by concerned citizens, with the majority of issues being related to parking enforcement (221), code violations (187), and traffic signals (135), 31.4% of the reported issues have no associated category. We provide two possible explanations of the high percentage of uncategorized reports: (i) category choices are tailored by SeeClickFix to the physical address provided by the user, and (ii) Albany authorities began their partnership with SeeClickFix in May 2013 leading to the introduction of individual categories in the beginning of 2014; before 2014, all issues reported are uncategorized.

We used labels to: (i) assess supervised classification models (i.e., machine learning models trained on the already manually classified data to learn to distinguish between issue types), and (ii) evaluate the performance of the proposed approach on previously "unseen" issues. We considered features directly extracted from issues' title, description, and tags. These intuitively capture the users' intent to categorize an issue using the SeeClickFix portal website or mobile app. We tokenized sentences into unigrams, removed punctuation (e.g., periods, commas, and apostrophes), stopwords (e.g., "a", "the", and "there"), and digits (e.g., "8th" and "31st"), and stemmed each word to its root (e.g., replace "parked" with "park"). Feature values correspond to the number of times a specific word or tag appears in the issue report. We excluded words present in $\geq 95\%$ and $\leq 2\%$ of all issues, respectively.

[1]https://seeclickfix.com/albany-county

[2]https://www.albany.edu/~dz973423/projects/nsf-scc-2017/

III. Problem Description

In this section, we formalize the problem of automatically inferring an issue's type from participatory reports with high accuracy while accounting for the effort of the framework in improving its chances of reaching highly accurate conclusion. We describe our model and define our optimization function.

A. Description

We consider a set of issues $\mathcal{I}$ (i.e., issues reported by citizens on SeeClickFix), were each issue $i \in \mathcal{I}$ has been reported by a user $u \in \mathcal{U}$, and has an associated title, description, tags, and media object (i.e., photo), along with a set of comments from users in $\mathcal{U}$. Each issue i is described by a vector of features $f(i) = \{y_1, y_2, \ldots, y_K\}$, where K is the total number of features, and $y_k \in \mathcal{Y}$. For illustration purposes, we assume that each issue i may belong to one of two hypotheses: $H_{\mathcal{C}_1}$, which denotes the true hypothesis that i is of type $\mathcal{C}_1$, or $H_{\mathcal{C}_2}$, where i is of type $\mathcal{C}_2$. We pose the challenge of automatic determination of the type of each reported issue as a sequential hypothesis testing problem and use an additive feature score to encode the belief that i is an instance of one class versus another.

For each feature y_n, the probability $p(y_n|H_{\mathcal{C}_1})$ (similarly $p(y_n|H_{\mathcal{C}_2})$) of the evaluation of the nth feature to observe value y_n when the true hypothesis is $H_{\mathcal{C}_1}$ (similarly for true hypothesis $H_{\mathcal{C}_2}$) is empirically computed from training data. The *a priori* probability $P(H_{\mathcal{C}_1}) = p$ of i being an instance of $\mathcal{C}_1$ is also estimated empirically. The probability of i being an instance of $\mathcal{C}_2$ can be computed as $P(H_{\mathcal{C}_2}) = 1 - p$.

To calculate the belief for i, the framework evaluates features sequentially as illustrated in Fig. 1. At each step, the framework has to select between stopping and continuing the evaluation process based on the accumulated information thus far and the cost of reviewing additional features. The cost coefficient $c_n > 0$, where $n = 1, \ldots, K$ represents the value of time and effort spent evaluating the nth feature. We also consider misclassification costs $M_{mj} \geqslant 0, m = \mathcal{C}_1, \mathcal{C}_2, j = 1, \ldots, L$, where M_{mj} denotes the cost of selecting type j when the true hypothesis is H_m, and L denotes the number of decision choices (e.g., $\mathcal{C}_1$ and $\mathcal{C}_2$). We factor misclassification costs into our approach to quantify the relative importance of detection errors. Note that a model that includes costs may not produce fewer errors than one that does not, and may not rank any higher in terms of overall accuracy, but it is likely to perform better in practical terms because it has a built–in bias in favor of less expensive errors towards one class versus another.

We now formally describe our proposed sequential evaluation process to minimize the number of features used to

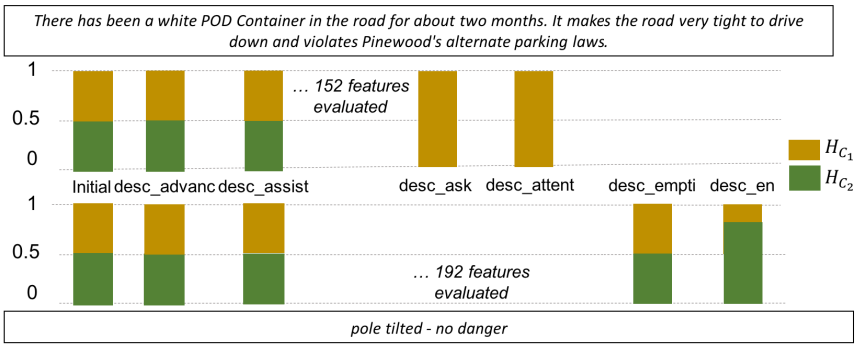

Fig. 1: Sample posterior probability evolution for parking enforcement (upper plot) and signs (lower plot) issues as more features are extracted and evaluated.

accurately classify issue i. Specifically, our proposed sequential evaluation process comprises a pair (R, D_R) of random variables. Random variable R (referred to as *stopping time* in decision theory) takes values in the set $\{0, \ldots, K\}$, and indicates the feature that the framework stops at. Random variable D_R denotes the possibility to select among L choices. It depends on R and takes values in the set $\{1, \ldots, L\}$. As an example, when $L = 3$, $D_R = 1$ corresponds to "$\mathcal{C}_1$ issue", $D_R = 2$ denotes "$\mathcal{C}_2$ issue", and $D_R = 3$ indicates "human expert inspection required". Assuming that the random variables y_n are *independent under each hypothesis* $H_m, m = \{\mathcal{C}_1, \mathcal{C}_2\}$, the conditional joint probability of $\{y_1, \ldots, y_n\}$ is given as $P(y_1, \ldots, y_n|H_m) = \prod_{k=1}^{n} p(y_k|H_m)$. Both the decision to stop at stage n (i.e., the event $\{R = n\}$), and the selection of possibility j (i.e., $D_R = j$) depend only on the accumulated information $\{y_1, \ldots, y_R\}$ by the stopping time R. Equivalently, features that may be examined in the future are not used.

B. Optimization Setup

To minimize the number of features considered for classifying issues without sacrificing accuracy, the stopping time R and the classification rule D_R have to be selected. To this end, we first define the following cost function:

$$J(R, D_R) = \mathbb{E}\Bigg\{ \sum_{n=1}^{R} c_n + \sum_{j=1}^{L} \sum_{m=\mathcal{C}_1,\mathcal{C}_2} M_{mj} P(D_R = j, H_m) \Bigg\}. \tag{1}$$

The first expression in the cost function regularizes the number of features, whereas the second expression penalizes the average cost of our classification rule. Our goal can be interpreted as finding the minimum average cost with respect to both random variables R and D_R, i.e., $\min\limits_{R, D_R} J(R, D_R)$, to derive the optimal stopping and classification rules. To prove that the optimal rule is to stop at corresponding stopping time R, we must first show how to obtain the optimum classification rule D_R for any given stopping time R. Once the optimal classification rule has been established, the resulting cost becomes only a function of R, and can thus be optimized with respect to R. Since D_R depends only on the accumulated information $\{y_1, \ldots, y_R\}$ by stopping time R, the *a posteriori probability* $\pi_n \triangleq P(H_{C_1}|y_1, \ldots, y_n)$, which corresponds to a sufficient statistic of the accumulated information, must be updated as more features are extracted and evaluated. Lemma 1 shows how to compute π_n iteratively.

Lemma 1. *The posterior probability π_n when the nth feature is evaluated to generate outcome y_n, and $\pi_0 = p$, is:*

$$\pi_n = \frac{p(y_n|H_{\mathcal{C}_1})\pi_{n-1}}{\pi_{n-1}p(y_n|H_{\mathcal{C}_1}) + (1 - \pi_{n-1})p(y_n|H_{\mathcal{C}_2})}, \tag{2}$$

Using Lemma 1 and the fact that $x_R = \sum_{n=0}^{K} x_n \mathbb{1}_{\{R=n\}}$ for any sequence of random variables $\{x_n\}$, where $\mathbb{1}_A$ is the indicator function for event A (*i.e.,* $\mathbb{1}_A = 1$ when A occurs, and $\mathbb{1}_A = 0$ otherwise), the average cost in Eq. (1) can be written compactly as:

$$J(R, D_R) = \mathbb{E}\Bigg\{ \sum_{n=1}^{R} c_n \Bigg\} + \mathbb{E}\Bigg\{ \sum_{j=1}^{L} \big(M_{\mathcal{C}_1 j}\pi_R + M_{\mathcal{C}_2 j}(1 - \pi_R) \big) \mathbb{1}_{\{D_R = j\}} \Bigg\}. \tag{3}$$

IV. Optimal Strategies

A. Classification Strategy

In order to obtain the optimal classification rule D_R for any stopping time R, an independent of D_R lower bound for the second part of Eq. (3) is needed. Since D_R contributes only to this portion of the average cost, the optimal classification rule D_R for a given stopping time R can then be derived. Theorem 2 provides such bound.

Theorem 2. *For any classification rule D_R given stopping time R, $\sum_{j=1}^{L} \big(M_{\mathcal{C}_1 j}\pi_R + M_{\mathcal{C}_2 j}(1 - \pi_R) \big) \mathbb{1}_{\{D_R = j\}} \geqslant g(\pi_R)$, where $g(\pi_R) \triangleq \min_{1 \leqslant j \leqslant L} \big[M_{\mathcal{C}_1 j}\pi_R + M_{\mathcal{C}_2 j}(1 - \pi_R) \big]$. The optimal rule is defined as follows:*

$$\mathcal{D}_R^{optimal} = \arg\min\nolimits_{1 \leqslant j \leqslant L} \big[M_{\mathcal{C}_1 j}\pi_R + M_{\mathcal{C}_2 j}(1 - \pi_R) \big]. \tag{4}$$

From Theorem 2, $J(R, D_R^{optimal}) \leqslant J(R, D_R)$, since the optimal classification rule results to the smallest average cost. Based on the this fact, Eq. (3) can be written as follows:

$$\widetilde{J} \triangleq J(R, D_R^{optimal}) = \min_{D_R} J(R, D_R) = \mathbb{E}\Bigg\{ \sum_{n=1}^{R} c_n + g(\pi_R) \Bigg\}, \tag{5}$$

which depends only on the stopping time R.

B. Stopping Strategy

The solution for optimizing $\widetilde{J}$ with respect to R can be determined by solving the optimization problem:

$$\min_{R\geqslant 0} \widetilde{J}(R) = \min_{R\geqslant 0} \mathbb{E}\Bigg\{ \sum_{n=1}^{R} c_n + g(\pi_R) \Bigg\}, \tag{6}$$

which constitutes a classical problem in optimal stopping theory for Markov processes [13]. We derive our optimal stopping strategy as described in Theorem 3 based on the observation that (i) the optimum strategy will consist of a maximum of $K+1$ stages since $R \in \{0, 1, \ldots, K\}$, and (ii) the solution we seek must also be optimum, if instead of the first stage we start from any intermediate stage and continue toward the final stage [14].

Theorem 3. *For $n = K-1, \ldots, 0$, the function $\bar{J}_n(\pi_n)$ is related to $\bar{J}_{n+1}(\pi_{n+1})$ through the equation:*

$$\bar{J}_n(\pi_n) = \min \Bigg[g(\pi_n), c_{n+1} + \sum_{y_{n+1}} A_n(y_{n+1}) \times \bar{J}_{n+1}\left(\frac{p(y_{n+1}|H_{\mathcal{C}_1})\pi_n}{A_n(y_{n+1})} \right) \Bigg], \tag{7}$$

where $A_n(y_{n+1}) \triangleq \pi_n p(y_{n+1}|H_{\mathcal{C}_1}) + (1-\pi_n) p(y_{n+1}|H_{\mathcal{C}_2})$ and $\bar{J}_K(\pi_K) = g(\pi_K)$.

The optimal stopping strategy derived by Eq. (7) has a very intuitive structure, i.e., stop at the stage where the cost of stopping (the first expression in the minimization) is no greater than the expected cost of continuing given all information accumulated at the current stage (the second expression in the minimization). Specifically, at each stage n, our method faces two options given π_n: (i) stop evaluating features and select optimally between the L possibilities, or (ii) continue and evaluate the next feature. The cost of stopping is $g(\pi_n)$, whereas the cost of continuing is $c_{n+1} + \sum_{y_{n+1}} A_n(y_{n+1}) \bar{J}_{n+1}\left(\frac{p(y_{n+1}|H_{\mathcal{C}_1})\pi_n}{A_n(y_{n+1})} \right)$.

C. Practical Considerations and Implementation

In this section, we describe ACTION, a novel algorithm for Automatic classification of civil issue reports with optimal online feature selection based on Theorems 2 and 3. Initially, the posterior probability π_0 is set to the prior probability p of an issue being an instance of type $\mathcal{C}_1$, and the two terms in Eq. (7) are compared. If they are equal, ACTION stops and classifies the issue based on the optimal rule of Eq. (4). Otherwise, the first feature is evaluated. ACTION repeats these steps until either it decides to stop, at which case it classifies the issue using $< K$ features, or all features are evaluated, in which case the issue is classified using all K features.

Note that the K functions $\bar{J}_n(\pi_n), n = 0, 1, \ldots, K-1$, are calculated using Eq. (7) by quantizing the interval $[0, 1]$ and computing the corresponding values. This computation relies only on *a priori* information to produce a $K \times d$ matrix, where each row corresponds to the value of the $\bar{J}_n(\cdot)$ function for different values of $\pi_n \in [0, 1]$. This computation needs to be performed only once and can be pre–calculated. Furthermore, probabilities $p(y_n|H_{\mathcal{C}_1}), p(y_n|H_{\mathcal{C}_2}), n = 1, \ldots, K, y_n \in \mathbb{Z}_{\geqslant 0}$, are empirically estimated from training data as $\hat{p}(y_n|H_{\mathcal{C}_1}) = \frac{N(y_n, \mathcal{C}_1)}{\sum_{y'_n} N(y'_n, \mathcal{C}_1)}$ and $\hat{p}(y_n|H_{\mathcal{C}_2}) = \frac{N(y_n, \mathcal{C}_2)}{\sum_{y'_n} N(y'_n, \mathcal{C}_2)}$, where $N(y_n, \mathcal{C}_1)$ and $N(y_n, \mathcal{C}_2)$ denote the number of issues that give rise to outcome y_n after extracting and evaluating the $n$the feature and constitute $\mathcal{C}_1$ and $\mathcal{C}_2$ issues, respectively. We also estimate the *a priori* probabilities as $[P(H_{\mathcal{C}_1}), P(H_{\mathcal{C}_2})]^T = [p, 1-p]^T = \left[\frac{N_{\mathcal{C}_1}}{N_{\mathcal{C}_1} + N_{\mathcal{C}_2}}, \frac{N_{\mathcal{C}_2}}{N_{\mathcal{C}_1} + N_{\mathcal{C}_2}} \right]^T$, where $N_{\mathcal{C}_1}$ and $N_{\mathcal{C}_2}$ denote the number of issues in the training set that constitute $\mathcal{C}_1$ and $\mathcal{C}_2$ issues, respectively. Hence the complexity of calculating $\bar{J}_n(\pi_n)$ is independent from the actual number of issues, which can be huge.

Finally, the ordering of features is crucial to the computation of the optimum average cost $\bar{J}_0(\pi_0)$. Consider for example the case of two features $f(i) = \{y_1, y_2\}$, where y_1 is the number of appearances of keyword "sign", and y_2 is the number of tags in an issue. The appearance of the type name (e.g., sign) in the title of an issue would intuitively discriminate issues better than the number of tags. Thus, if feature y_2 was to be examined first, it would be very probable for feature y_1 to be examined as well to improve the chances of accurate classification. Alternatively, if y_1 was to be evaluated first, our framework could reach a decision using one feature only. To avoid the computational complexity of evaluating all $K!$ possible feature orderings, we sort features in increasing order of $c_n(\epsilon_{\mathcal{C}_1} + \epsilon_{\mathcal{C}_2})$ to promote low cost (i.e., c_n) features that at the same time are expected to result in few errors (i.e., $\epsilon_{\mathcal{C}_1} + \epsilon_{\mathcal{C}_2}$). To implement this heuristic, we find where the mass of observations lie for each issue type, and sum the probabilities that are left out. Our framework can be easily extended to accommodate other heuristics.

V. Numerical Results

In this section, we present the evaluation of ACTION and compare its performance to (i) a linear SVM classifier [10], and (ii) a standard Bayesian detection approach [15] that uses all features. In our experiments, L is set to 2 (i.e., issues may belong to one of two categories), varying feature costs $c_n \in \{0, 0.00001, 0.0001, 0.001, 0.1, 0.2, 0.3, 0.5, 0.7, 1, 10\}$ and misclassification costs $M_{\mathcal{C}_1 2}, M_{\mathcal{C}_2 1} \in \{1, 1.5, 5, 10, 25, 50\}$ are considered, and five–fold cross validation results are reported. To test the robustness of ACTION, we experimented with three overlapping scenarios of closely related categories: (i) "Code Violations" and "Signs", (ii) "Parking Enforcement" and "Code Violations", and (iii) "Parking Enforcement" and "Signs". In each case, a balanced training (and testing) dataset is created comprising ~ 300 issues from each corresponding type, whereas the total number of features considered surpasses 1K ($1,286$, $1,064$, and $1,111$, accordingly).

Fig. 2 illustrates the error probability achieved by ACTION as the average number of features used by the algorithm increases. Results are reported for constant misclassification costs (i.e., $M_{\mathcal{C}_1 1} = M_{\mathcal{C}_2 2} = 0$ and $M_{\mathcal{C}_1 2} = M_{\mathcal{C}_2 1} = 1$) and for varying values of c_n when all features have the same cost

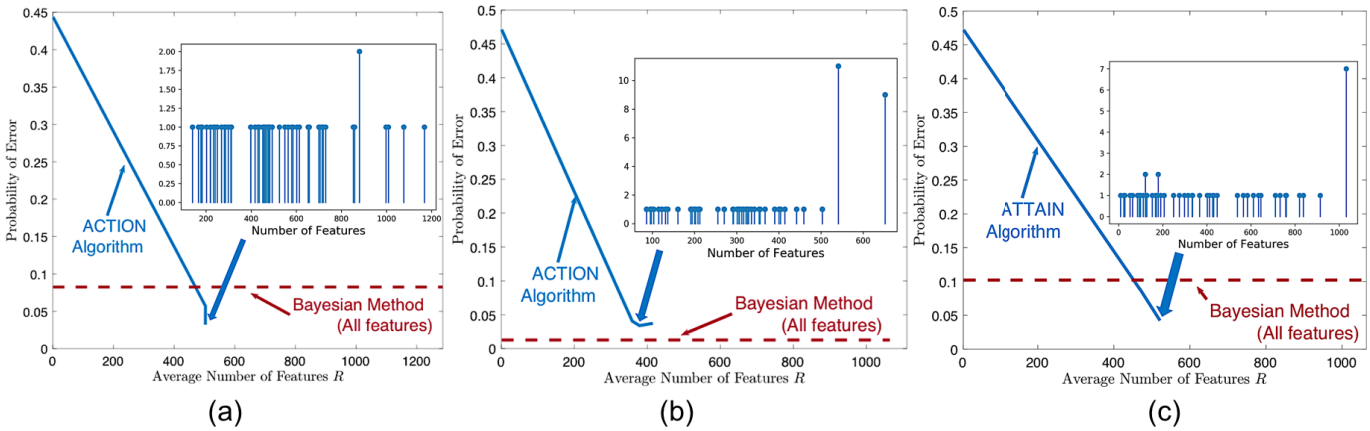

Fig. 2: Probability of error as a function of the expected number of features for (a) "Code Violations" and "Signs", (b) "Parking Enforcement" and "Code Violations", and (c) "Parking Enforcement" and "Signs", respectively. Insets show the distribution of number of features used by ACTION to classify issues in each scenario in the case of the smaller error probability.

(i.e., $c_n = c$). The insets in Fig. 2 show the number of features used by ACTION to classify issues in the testing dataset for an average number of (a) ~ 508.45, (b) ~ 380.64, and (c) ~ 434.25 features, accordingly. For comparison, the error probability achieved by a standard Bayesian method that uses all available features is also included in the figure. The SVM and Bayesian methods achieve 98% and 93% accuracy on average, using however $\sim$ twice as many features as compared to ACTION. We note that in two out of the three scenarios (i.e., "Code Violations" and "Signs", "Parking Enforcement" and "Signs") ACTION attains better accuracy than the Bayesian method using at least 41% less features. This is because the performance of the Bayesian method is inversely impacted by the highly noisy features (e.g., similar keywords describing parking and signs) in our dataset. As expected, when the average number of features used is small, ACTION exhibits large error probability. However, as this number increases, performance improves dramatically. In all cases, ACTION achieves $\sim 96\%$ accuracy when $c_n \in \{0, 0.00001, 0.0001, 0.001\}$; ACTION achieves best accuracy using ~ 500 features in each case. This corresponds to at least 50% reduction on average in the number of features used while at the same time significantly improving the overall classification accuracy. Different values of costs c_n and misclassification costs $M_{c_1 2}$ and $M_{c_2 1}$ result in different error probability values, while trading–off false alarm and misdetection probabilities.

VI. Conclusions and Future Work

In this paper, a sequential hypothesis testing formulation was proposed to address the problem of automatic classification of civil issue requests on an online community advocacy platform. An optimization function was defined in terms of the cost of features and the average cost of the classification strategy, and the optimal solution was determined. The proposed algorithm that implements the optimal solution achieves at least 50% reduction on the average number of features used to reach a classification decision. In future work, we plan to extend our framework so as to exploit the multitude of available features.

References

[1] J. A. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava, "Participatory sensing," 2006.

[2] P. Dutta, P. M. Aoki, N. Kumar, A. Mainwaring, C. Myers, W. Willett, and A. Woodruff, "Common sense: participatory urban sensing using a network of handheld air quality monitors," in *Proceedings of the 7th ACM conference on embedded networked sensor systems*. ACM, 2009, pp. 349–350.

[3] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher, "Greengps: a participatory sensing fuel-efficient maps application," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 2010, pp. 151–164.

[4] P. Zhou, Y. Zheng, and M. Li, "How long to wait? predicting bus arrival time with mobile phone based participatory sensing," *IEEE Transactions on Mobile Computing*, vol. 13, no. 6, pp. 1228–1241, 2014.

[5] L. Deng and L. P. Cox, "Livecompare: grocery bargain hunting through participatory sensing," in *Proceedings of the 10th workshop on Mobile Computing Systems and Applications*. ACM, 2009, p. 4.

[6] S. F. King and P. Brown, "Fix my street or else: using the internet to voice local public service concerns," in *Proceedings of the 1st international conference on Theory and practice of electronic governance*. ACM, 2007, pp. 72–80.

[7] I. Mergel, "Distributed democracy: Seeclickfix. com for crowdsourced issue reporting," 2012.

[8] A. L. Kavanaugh, E. A. Fox, S. D. Sheetz, S. Yang, L. T. Li, D. J. Shoemaker, A. Natsev, and L. Xie, "Social media use by government: From the routine to the critical," *Government Information Quarterly*, vol. 29, no. 4, pp. 480–491, 2012.

[9] D. C. Brabham, "A model for leveraging online communities," *The participatory cultures handbook*, vol. 120, 2012.

[10] N. Beck, "Classification of issues in the public space using their textual description and geo-location."

[11] C. Masdeval and A. Veloso, "Mining citizen emotions to estimate the urgency of urban issues," *Information systems*, vol. 54, pp. 147–155, 2015.

[12] M. Budde, J. D. M. Borges, S. Tomov, T. Riedel, and M. Beigl, "Improving participatory urban infrastructure monitoring through spatio-temporal analytics," in *3rd ACM SIGKDD International Workshop on Urban Computing*. ACM, 2014.

[13] A. N. Shiryaev, *Optimal Stopping Rules*. Springer Science & Business Media, 2007, vol. 8.

[14] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 2005, vol. 1.

[15] H. L. Van Trees, *Detection, estimation, and modulation theory, part I: detection, estimation, and linear modulation theory*. John Wiley & Sons, 2004.