ON–THE–FLY FEATURE SELECTION AND CLASSIFICATION WITH APPLICATION TO CIVIC ENGAGEMENT PLATFORMS

Yasitha Warahena Liyanage[⋆] *Daphney–Stavroula Zois*[⋆] *Charalampos Chelmis*[†]

[⋆]Electrical and Computer Engineering Department
[†]Computer Science Department
University at Albany, SUNY, Albany, NY, USA
Emails: {yliyanage, dzois, cchelmis}@albany.edu

ABSTRACT

Online feature selection and classification is crucial for time sensitive decision making. Existing work however either assumes that features are independent or produces a fixed number of features for classification. Instead, we propose an optimal framework to perform joint feature selection and classification on–the–fly while relaxing the assumption on feature independence. The effectiveness of the proposed approach is showed by classifying urban issue reports on the SeeClickFix civic engagement platform. A significant reduction in the average number of features used is observed without a drop in the classification accuracy.

Index Terms— Government 2.0, quickest detection, Bayesian networks, correlated features, Markov blanket

1. INTRODUCTION

In the heart of supervised machine learning lies feature selection, the goal of which is to choose a subset of features from a larger set of potentially redundant features so as to maximize classification accuracy [1]. In applications (e.g., [2–4]), where time sensitive and accurate decision making is essential, feature selection and/or classification need often to be carried out in a streaming fashion.

In this paper, the problem of on–the–fly feature selection and classification is considered. Specifically, we propose a method that utilizes a varying number of features to perform joint feature selection and classification of data instances as they become available. This is in stark contrast to popular offline and online feature selection and dimensionality reduction methods [1, 5–9] that identify a subset of discriminative features, common to all instances for classification. To this end, we define an optimization problem which simultaneously minimizes the number of features evaluated and maximizes classification accuracy, the solution of which leads to an approach that sequentially reviews features to classify a data instance once it determines that including additional features cannot improve the quality of classification. Since feature ordering plays a critical role in this sequential evaluation process, we introduce a novel feature ordering method by utilizing the Markov blanket criterion [10]. Specifically, we model dependencies among features using a Bayesian network, and order features such that each selected feature contains the maximum possible new information about the class/target variable with respect to the already evaluated feature set.

Our motivation stems from the problem of urban issue reports classification in civic engagement platforms [11, 12]. Such platforms have recently emerged to facilitate the communication between concerned citizens and the government by providing the former with the means to electronically report non–emergency urban issues (e.g. potholes or noise complaints) [12, 13]. To ensure the continuous engagement and participation of citizens, urban issue reports need to be timely addressed by their local governments. Prior work on classification of urban issue reports in civic engagement platforms has mostly focused on either binary classification of reports into categories [14–16] or importance [17, 18], and typically need large training datasets to achieve good accuracy [17, 19]. All such methods assume features to be independent. On the other hand, existing feature selection methods [5–9] produce a fixed set of features to be used during classification; we have shown in our prior work [20] that this approach is sub–optimal.

2. PROBLEM DESCRIPTION

Consider a set $\mathcal{S}$ of data instances, with each data instance $s \in \mathcal{S}$ being described using an assignment of values $f \triangleq \{f_1, f_2, \ldots, f_K\}$ to a set $F \triangleq \{F_1, F_2, \ldots, F_K\}$ of features. Each data instance s is drawn from some probability distribution over the feature space such that for each assignment f to F, the probability $P(F = f)$ is non zero. Further, each instance s may belong to one of L possible classes, with a

This material is based upon work supported by the National Science Foundation under Grant No. ECCS–1737443.

corresponding *a priori* probability $P(\mathcal{C} = \mathcal{C}_i) = p_i$ for each assignment $\mathcal{C}_i, i = 1, 2, \ldots, L$, of the class variable $\mathcal{C}$. Finally, coefficients $e_n, n = 1, 2, \ldots, K$, denote the cost of evaluating feature F_n and misclassification cost Q_{ij} represents the cost of selecting class $\mathcal{C}_j$ when class $\mathcal{C}_i$ is true, where $i, j = 1, 2, \ldots, L$.

In contrast to our prior work [16, 18, 20] we assume features to be dependent, both among themselves, as well as with the class variable. We use a Bayesian network to model such dependencies [10]. Specifically, we consider a Bayesian network $\mathcal{B} \triangleq \{G, \Theta\}$, where G denotes a directed acyclic graph whose nodes correspond to features in F and the class variable $\mathcal{C}$, edges correspond to direct dependencies among such variables, and Θ represents the set of parameters, (i.e., conditional probability distributions) that characterize the network (see Figure 1). It is important to note that learning such a Bayesian network is beyond the scope of this paper. Well–known methods for this task exist [21, 22], thus, we assume that the Bayesian network is given and its parameters are learned during training.

The goal is to speedup the classification process by considering the dependency structure among features and choosing the least number of most informative features that will maximize the classification accuracy in an online fashion. Specifically, in order to select one out of L possible classes for each instance s, the proposed approach evaluates features sequentially by choosing the features that are: (i) highly correlated with the class variable, and (ii) conditionally independent with the already evaluated feature set. At each step, our approach considers the cost of examining the remaining features to decide between continuing the process or if enough information is available for a classification decision to be reached. Herein, we introduce a pair of random variables (R, D_R), where R is an instance of set $\{0, \ldots, K\}$, while random variable D_R that depends on R, is an instance of set $\{1, \ldots, L\}$. Specifically, R denotes the feature at which the framework stops at, where the event $\{R = n\}$ depends only on feature values up to feature F_n. On the other hand, D_R denotes the possibility to select among L classes, where the event $\{D_R = m\}$ represents choosing class $\mathcal{C}_m$ based on the information accumulated up to feature R.

In order to find variables R and D_R, our approach minimizes the following cost function:

$$J(R, D_R) = \mathbb{E}\left\{ \sum_{n=1}^{R} e_n \right\} + \sum_{j=1}^{L} \sum_{i=1}^{L} Q_{ij} P(D_R = j, \mathcal{C}_i), \tag{1}$$

where the first term represents the cost of evaluating features, while the second term penalizes the misclassification cost.

3. OPTIMUM SOLUTION

In this section, we derive optimal strategies to solve the optimization problem defined in Section 2. The first step is to extract a highly correlated feature subset with the class variable $\mathcal{C}$ from the Bayesian network. To achieve this, we extract the Markov blanket $M_{\mathcal{C}}$ from the Bayesian network.

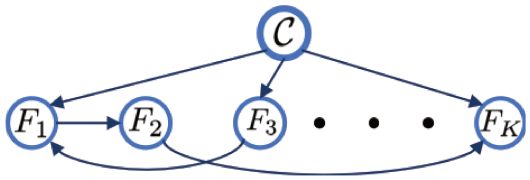

Fig. 1. Sample dependency structure of feature set F and class variable $\mathcal{C}$ using a Bayesian Network.

Definition 1. *Markov Blanket [10]: The set $M_{\mathcal{C}} \subseteq F$ is a Markov blanket for class variable $\mathcal{C}$, if $\mathcal{C}$ is conditionally independent of the set $\{F - M_{\mathcal{C}}\}$ given $M_{\mathcal{C}}$.*

This implies that any feature $F_i \in \{F - M_{\mathcal{C}}\}$ provides no additional information about the class variable $\mathcal{C}$ beyond what is already available in $M_{\mathcal{C}}$. In other words, Markov blanket $M_{\mathcal{C}}$ provides sufficient information to predict $\mathcal{C}$ [10]. In the rest of the paper, we focus on the sub–network $\mathcal{B}_{\mathcal{C}}$, which contains only the nodes corresponding to features in $M_{\mathcal{C}}$ and the node corresponding to $\mathcal{C}$.

The next step is to evaluate features in the set $M_{\mathcal{C}}$ sequentially. We consider the *a posteriori probability* vector π_n defined as follows:

$$\pi_n \triangleq [\pi_n^1, \pi_n^2, \ldots, \pi_n^L]^T, \tag{2}$$

where $\pi_n^i = P(\mathcal{C}_i | F_1, \ldots, F_n)$. To simplify the notation, $P(\mathcal{C}_i | F_1, \ldots, F_n)$ is used in lieu of $P(\mathcal{C} = \mathcal{C}_i | F_1 = f_1, \ldots, F_n = f_n)$ hereafter. This sufficient statistic can be computed recursively as indicated in Lemma 1.

Lemma 1. *The a posteriori probability vector π_n, where the nth feature F_n is evaluated to generate outcome f_n, is given by the following expression:*

$$\pi_n = \frac{\text{diag}\left(\Delta_n(F_n | F_1, \ldots, F_{n-1}, \mathcal{C})\right)\pi_{n-1}}{\Delta_n^T(F_n | F_1, \ldots, F_{n-1}, \mathcal{C})\pi_{n-1}}, \tag{3}$$

where $\Delta_n(F_n | F_1, \ldots, F_{n-1}, \mathcal{C}) = [P(F_n | F_1, \ldots, F_{n-1}, \mathcal{C}_1), \ldots, P(F_n | F_1, \ldots, F_{n-1}, \mathcal{C}_L)]^T$, $\text{diag}(A)$ denotes a diagonal matrix with diagonal elements being the elements in vector A, and $\pi_0 = [p_1, p_2, \ldots, p_L]^T$.

To accommodate early stopping, the recursion should be carried out so that once $n - 1$ features are evaluated, the nth feature is chosen such that it contains the maximum possible new information about the class variable with respect to the already evaluated feature set. We begin with the feature $F_i \in M_{\mathcal{C}}$ that has the highest correlation/mutual information with $\mathcal{C}$. Without loss of generality, let us call this F_1. Consider the portion of the Markov blanket $\widetilde{M}_1 \subseteq \{M_{\mathcal{C}} \cup \mathcal{C}\}$

of feature F_1 inside the sub–network $\mathcal{B}_{\mathcal{C}}$. More precisely, $\widetilde{M}_1 \triangleq M_1 \cap \{M_{\mathcal{C}} \cup \mathcal{C}\}$, where $M_1 \subseteq F$ is the Markov blanket of F_1. According to Definition 1, feature F_1 is conditionally independent of any feature $F_i \in \{M_{\mathcal{C}} - \widetilde{M}_1\}$ given $\widetilde{M}_1$. In other words, $\widetilde{M}_1$ subsumes all information that F_1 has about $\{M_{\mathcal{C}} - \widetilde{M}_1\}$. Hence, F_2 should be selected from the set $\{M_{\mathcal{C}} - \widetilde{M}_1\}$, such that it contains the highest mutual information with $\mathcal{C}$. Similarly, once $n-1$ features are evaluated, the nth feature F_n should be selected from the set $\{M_{\mathcal{C}} - \{\widetilde{M}_1 \cup \widetilde{M}_2 \cup \cdots \cup \widetilde{M}_{n-1}\}\}$ such that it contains the highest mutual information with $\mathcal{C}$. Every time a new feature is selected, the *a posteriori probability* vector needs to be updated using the probability vector $\Delta_{n+1}^T(F_n|F_1,\ldots,F_{n-1},\mathcal{C})$ (see Lemma 1). This marginal distribution can be computed using exact inference algorithms (e.g. belief propagation) [10, 22].

Lemma 2. *Based on the fact that $y_R = \sum_{n=0}^{K} y_n \mathbb{1}_{\{R=n\}}$ for any sequence of random variables $\{y_n\}$, 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 probability $P(D_R = j, \mathcal{C}_i)$ can be written as $P(D_R = j, \mathcal{C}_i) = \mathbb{E}\left\{\pi_R^i \mathbb{1}_{\{D_R=j\}}\right\}$.*

Using Lemma 2, the average cost in Eq. (1) can be written compactly as:

$$J(R, D_R) = \mathbb{E}\left\{\sum_{n=1}^{R} e_n + \sum_{j=1}^{L}\left(\sum_{i=1}^{L} Q_{ij}\pi_R^i\right)\mathbb{1}_{\{D_R=j\}}\right\}. \quad (4)$$

Note that we can rewrite the average cost in Eq. (4) using the *a posteriori probability* vector π_n as follows:

$$J(R, D_R) = \mathbb{E}\left\{\sum_{n=1}^{R} e_n + \sum_{j=1}^{L} Q_j^T \pi_R \mathbb{1}_{\{D_R=j\}}\right\}, \quad (5)$$

where $Q_j \triangleq [Q_{1,j}, Q_{2,j}, \ldots, Q_{L,j}]^T$.

To obtain R, we must first obtain the optimal decision rule D_R for any given R. In the process of finding the optimal decision, we need to find a lower bound (independent of D_R) for the second term inside the expectation in Eq. (5), which is the part of the equation that depends on D_R. Theorem 3 provides such a bound.

Theorem 3. *For any classification rule D_R given R, $\sum_{j=1}^{L} Q_j^T \pi_R \mathbb{1}_{\{D_R=j\}} \geqslant g(\pi_R)$, where $g(\pi_R) \triangleq \min_{1\leqslant j\leqslant L}\left[Q_j^T \pi_R\right]$. The optimal rule is defined as follows:*

$$D_R^{optimal} = \arg\min_{1\leqslant j\leqslant L}\left[Q_j^T \pi_R\right]. \quad (6)$$

From Theorem 3, we conclude that:

$$J(R, D_R) \geqslant J(R, D_R^{optimal}), \text{ where}$$
$$J(R, D_R^{optimal}) = \min_{D_R} J(R, D_R). \quad (7)$$

Thus, we can reduce the cost function in Eq. (5) to one which depends only on R as follows:

$$\widetilde{J}(R) = \mathbb{E}\left\{\sum_{n=1}^{R} e_n + g(\pi_R)\right\}. \quad (8)$$

To optimize the cost function in Eq. (8) with respect to R, we need to solve the following optimization problem:

$$\min_{R\geqslant 0} \widetilde{J}(R) = \min_{R\geqslant 0} \mathbb{E}\left\{\sum_{n=1}^{R} e_n + g(\pi_R)\right\}. \quad (9)$$

Since $R \in \{0, 1, \ldots, K\}$, the optimum strategy will consist of a maximum of $K+1$ stages, where the optimum scheme must minimize the corresponding average cost going from stages 0 to K. The solution can be obtained using *dynamic programming* [23].

Theorem 4. *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:*

$$\bar{J}_n(\pi_n) = \min\Bigg[g(\pi_n), e_{n+1} + \sum_{F_{n+1}} \Delta_{n+1}^T\big(F_{n+1}|F_1,\ldots,F_n, \mathcal{C}\big)\pi_n \times \bar{J}_{n+1}\left(\frac{\operatorname{diag}\big(\Delta_{n+1}(F_{n+1}|F_1,\ldots,F_n,\mathcal{C})\big)\pi_n}{\Delta_{n+1}^T(F_{n+1}|F_1,\ldots,F_n,\mathcal{C})\pi_n}\right)\Bigg], \quad (10)$$

where $\bar{J}_K(\pi_K) = g(\pi_K)$.

The optimal strategy derived from Eq. (10) has a very intuitive structure: it stops at stage n, where the cost of stopping (the first expression in the minimization) is no greater than the expected cost of continuing given all information accumulated up to the current stage n (the second expression in the minimization). Specifically, at each stage n, our method faces two options given π_n: (i) stop evaluating features and optimally selecting between the L classes, or (ii) continue with evaluating the next feature. We use value iteration to obtain the optimal stopping solution in Eq. (10), where we uniformly quantize the *a posteriori probability* space and iteratively update the functions $\bar{J}_n(\pi_n), n = 1, \ldots, K$, until convergence [24].

4. EXPERIMENTAL RESULTS

We illustrate the performance of our approach on a real–world dataset of $2,195$ issues, spanning a time period between Jan 5, 2010 and Feb 10, 2018, for the capital of the state of New York, collected from SeeClickFix[1]. Without loss of generality, we consider a set of four issue types (4 classes), i.e., {Parking Enforcement, Code Violation, Traffic Signal Repair, Signs (missing, needed, or damaged)}. The goal is to assign

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

Table 1. Performance comparison with baselines.

Method		Acc.	Precision (Avg±Std)	Recall (Avg±Std)	Avg. # feat.
Our Approach	$c = 0.4$	0.412	0.312 ± 0.393	0.476 ± 0.478	1.0
	$c = 0.3$	0.824	0.807 ± 0.142	0.844 ± 0.145	2.072
	$c = 0.2$	0.939	0.935 ± 0.083	0.939 ± 0.055	2.293
	$\mathbf{c = 0.1}$	$\mathbf{0.943}$	$\mathbf{0.945 \pm 0.070}$	$\mathbf{0.942 \pm 0.055}$	$\mathbf{2.471}$
	$c = 0.0$	0.945	0.957 ± 0.053	0.940 ± 0.048	3.793
OFS-Density [7]		0.951	0.938 ± 0.072	0.951 ± 0.045	15.0
SAOLA [8]		0.741	0.765 ± 0.289	0.811 ± 0.307	5.20
OSFS [9]		0.735	0.708 ± 0.359	0.807 ± 0.321	3.80
FAST–OSFS [9]		0.701	0.611 ± 0.473	0.719 ± 0.482	4.0
ASSESS [20]		0.949	0.941 ± 0.068	0.949 ± 0.056	4.867
SVM–FS [19]		0.947	0.950 ± 0.074	0.947 ± 0.043	6
PCA		0.966	0.962 ± 0.038	0.968 ± 0.036	190
SVM–L		$\mathbf{0.970}$	0.961 ± 0.038	0.966 ± 0.035	1606
SVM–G		0.968	$\mathbf{0.964 \pm 0.036}$	$\mathbf{0.969 \pm 0.034}$	1606
RF (d=5)		0.947	0.941 ± 0.048	0.950 ± 0.056	1606
RF (d=10)		0.962	0.961 ± 0.037	0.961 ± 0.033	1606
XG–B		0.964	0.957 ± 0.048	0.962 ± 0.040	1606

each issue to one of the four classes, using a total of $1,606$ features, extracted from issues' title and description by tokenizing sentences into unigrams. A feature value corresponds to the number of appearances of a specific word in an issue.

To obtain the highly correlated feature set with the class variable, we filter out features based on a threshold α on the mutual information between each feature and the class variable. For ease of implementation, we consider a tree dependency structure, where the class variable is the root of the tree and each feature node contains the class variable and at most one other feature node as its parents. This type of model can be efficiently trained by computing pairwise conditional mutual information among features and building the maximum spanning tree [25]. We use a smoothed maximum likelihood estimator to estimate the conditional probability tables, (i.e., $P(F_n|\Pi_{F_n})$, where Π_{F_n} denotes the set of parents of F_n), after binning the feature space. For example, $\hat{P}(F_a = f_a|F_b = f_b, \mathcal{C} = \mathcal{C}_i) = \frac{N_{a,b,i}+1}{N_{b,i}+V}$, where $N_{a,b,i}$ denotes the number of samples that satisfy $F_a = f_a$ and $F_b = f_b$, and belong to class $\mathcal{C}_i$, ($N_{b,i}$ is defined in a similar way) and V is the number of bins considered. We estimate the *a priori* probabilities as $P(\mathcal{C}_i) = \frac{N_i}{\sum_{i=1}^{L} N_i}, i = 1, \ldots, L$. In our experiments, threshold α is set to 0.1, number of bins $V = 10$, $L = 4$ (i.e., 4 issue types), misclassification costs are set to $Q_{ij} = 1, \forall i \neq j$ and $Q_{ij} = 0, \forall i = j$, and feature costs $c_n \in \{0, 0.1, 0.2, 0.3, 0.4\}$ are considered.

We compare the performance of our approach to (i) online feature selection methods OFS-Density [7], SAOLA [8], OSFS [9], FAST–OSFS [9], (ii) our own prior work, ASSESS [20], (iii) offline feature selection and dimentionality reduction methods: SVM–FS [19], Principal Component Analysis with SVM classifier (PCA), and (v) state–of–the–art classifiers: Support Vector Machines with linear (SVM–L) and Gaussian (SVM–G) kernels, inherently multiclass classifiers, namely Random Forest (RF) with maximum tree depths $d = 5, 10$, and XG Boosting (XG–B). For online feature selection methods, we use KNN classifier to evaluate a selected feature subset. Five–fold cross validation results are reported.

In Table 1, we summarize the performance of our approach compared to all baselines. Among all baselines, SVM–L achieves the highest accuracy, SVM–G achieves the highest precision and recall, but requires $\sim$650 times more features than our approach for a mere 2.8%, 2.0% and 2.9% improvement in accuracy, precision and recall, respectively. OFS–Density [7] outperforms all other online feature selection algorithms. However, OFS–Density [7] requires $\sim$6 times as many features, while degrading 0.7% in precision, for a mere 0.8% and 1.0% improvement in accuracy and recall, respectively, compared to our approach. By relaxing the feature independence assumption in ASSESS [20], our approach reduces the number of features used by $\sim$50%, while incurring 0.6% and 0.7% degradation in accuracy and recall, respectively.

5. CONCLUSION

In this paper, we have addressed the problem of selecting the least number of most informative features per data instance for fast and accurate classification. An optimization problem was defined in terms of the cost of evaluating features and the Bayes risk associated with the classification rule, and its optimal solution was obtained. A novel feature ordering technique was introduced to accommodate early stopping of the classification task by taking feature dependencies into account. Evaluation on a real–world dataset demonstrated the ability of the proposed approach to reduce the number of features used by up to $\sim$50%, while maintaining classification performance as compared to the state–of–the–art. In our future work, we plan to simultaneously learn the feature dependency structure and find the optimal feature ordering on–the–fly.

6. REFERENCES

[1] Isabelle Guyon and André Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[2] Jae-Hyun Seo, Yong Hee Lee, and Yong-Hyuk Kim, "Feature selection for very short-term heavy rainfall prediction using evolutionary computation," *Advances in Meteorology*, vol. 2014, 2014.

[3] Su Yang, "On feature selection for traffic congestion prediction," *Transportation Research Part C: Emerging Technologies*, vol. 26, pp. 160–169, 2013.

[4] Ming-Chi Lee, "Using support vector machine with a hybrid feature selection method to the stock trend prediction," *Expert Systems with Applications*, vol. 36, no. 8, pp. 10896–10904, 2009.

[5] Girish Chandrashekar and Ferat Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[6] John P Cunningham and Zoubin Ghahramani, "Linear dimensionality reduction: Survey, insights, and generalizations," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 2859–2900, 2015.

[7] Peng Zhou, Xuegang Hu, Peipei Li, and Xindong Wu, "Ofs-density: A novel online streaming feature selection method," *Pattern Recognition*, vol. 86, pp. 48–61, 2019.

[8] Kui Yu, Xindong Wu, Wei Ding, and Jian Pei, "Towards scalable and accurate online feature selection for big data," in *2014 IEEE International Conference on Data Mining*. IEEE, 2014, pp. 660–669.

[9] Xindong Wu, Kui Yu, Wei Ding, Hao Wang, and Xingquan Zhu, "Online feature selection with streaming features," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 5, pp. 1178–1192, 2012.

[10] Judea Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1988.

[11] Soon Ae Chun, Stuart Shulman, Rodrigo Sandoval, and Eduard Hovy, "Government 2.0: Making connections between citizens, data and government," *Info. Pol.*, vol. 15, no. 1,2, pp. 1–9, Apr. 2010.

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

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

[14] Y. Sano, K. Yamaguchi, and T. Mine, "Category estimation of complaint reports about city park," in *2015 IIAI 4th International Congress on Advanced Applied Informatics*, July 2015, pp. 61–66.

[15] N. Beck, "Classification of Issues in the Public Space Using Their Textual Description and Geo-Location," .

[16] Daphney-Stavroula Zois, Christopher Yong, Charalampos Chelmis, Angeliki Kapodistria, and Wonhyung Lee, "Improving monitoring of participatory civil issue requests through optimal online classification," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2018, pp. 2034–2038.

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

[18] Yasitha Liyanage, Mengfan Yao, Christopher Yong, Daphney-Stavroula Zois, and Charalampos Chelmis, "What matters the most? optimal quick classification of urban issue reports by importance," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2018, pp. 106–110.

[19] Sachio Hirokawa, Takahiko Suzuki, and Tsunenori Mine, "Machine learning is better than human to satisfy decision by majority," in *Proceedings of the International Conference on Web Intelligence*, New York, NY, USA, 2017, WI '17, pp. 694–701, ACM.

[20] Yasitha Warahena Liyanage, Daphney-Stavroula Zois, Charalampos Chelmis, and Mengfan Yao, "Automating the classification of urban issue reports: an optimal stopping approach," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3137–3141.

[21] Richard E Neapolitan et al., *Learning bayesian networks*, vol. 38, Pearson Prentice Hall Upper Saddle River, NJ, 2004.

[22] Daphne Koller and Nir Friedman, *Probabilistic graphical models: principles and techniques*, MIT press, 2009.

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

[24] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.

[25] Nir Friedman and Moises Goldszmidt, "Building classifiers using bayesian networks," in *Proceedings of the national conference on artificial intelligence*, 1996, pp. 1277–1284.