Incomplete Label Uncertainty Estimation for Petition Victory Prediction with Dynamic Features

Junxiang Wang,Yuyang Gao,Andreas Züfle,Jingyuan Yang and Liang Zhao
George Mason University
{jwang40,ygao13,azufle,jyang53,lzhao9}@gmu.edu

***Abstract*— It is important for decision-makers to effectively and proactively differentiate the significance of various public concerns, and address them with optimal strategy under the limited resources. Online Petition Platforms (OPPs) are replacing traditional social and market surveys for the advantages of low financial cost and high-fidelity social indicators. Despite benefits from OPPs, the raw information from millions of petition signers can easily overwhelm decision makers. In addition, spatio-temporal and semantic dissemination patterns increase the complexity of such OPP data. These two aspects show the necessity of a framework that learns from all available data, which is encoded by dynamic representation of features, to predict whether a petition will successfully lead to a change by decision makers. To build such framework, we need to overcome several challenges including: 1) missing values in dynamic features; 2) strong uncertainty in petition prediction; 3) unknown labels for ongoing petitions and 4) Scalability regarding increasing features and petitions. To address these difficulties simultaneously, we propose a novel chain-structure Multi-task Learning framework with Uncertainty Estimation (MLUE) to predict potentially victorious petitions, which facilitates the process of decision making. Specifically, we divide data into different Increasing Feature Blocks (IFBs) according to missing patterns. Besides, we propose a novel criterion to estimate uncertainty in order to label petitions as early as possible. To handle the challenge of scalability, we present an Expectation-Maximization (EM)-based algorithm to optimize the non-convex objective function accurately and efficiently. Various experiments on six petition datasets demonstrate that our MLUE outperformed other baselines by a large margin.**

***Index Terms*—petition victory prediction, unlabeled data, missing value, multi-task learning, uncertainty estimation**

I. Introduction

The rise of Online Petition Platform (OPP), which is a form of web-based petition host, came out over the recent decades and spurred with the internet and social networking. For example, Change.org [1], which was founded in 2007, has owned over 190 million users and hundreds of daily petitions covering various social aspects from health, economics, to government policies by July 2017. These petition websites have grown to be an important way to detect and track timely public concerns toward societal issues, as well as a creative attempt to fill the gap between the increasing public concerns and the decision-makers' attention. One classic example of victorious online petitions which aimed at reducing serious food waste was achieved by making the decision-makers of Whole Foods Market agree on stopping discarding "ugly looking fruits"[2].

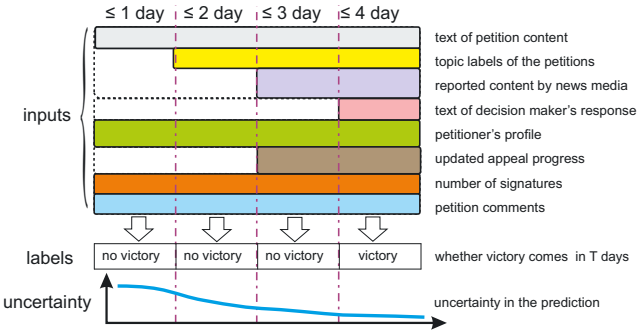


Fig. 1. An example of petition victory prediction task

Typically, an online petition can be easily created by using some web-based petition hosts to gather enough signatures in order to get the attention of the responsible decision-makers. A victorious petition means that the decision-makers take action to address the concern of the petition-launcher. Although the outstanding benefits of OPPs such as low financial cost and easy accessibility attract millions of users, this also make the online petitions suffer from two challenges: firstly, they are usually poorly organized so that massive nearly-duplicated and correlated petitions overwhelm the decision-makers; moreover, decision-makers are distracted by sophisticated problems of spatio-temporal and semantic dissemination from various similar petitions. There is a critical need for decision-makers to optimize the utilization of their limited resource and prioritize those petitions with higher victory probability more efficiently and proactively in order to minimize the social upheavals and uncertaintys.

However, developing such a petition victory prediction framework is a non-trivial task, which needs to tackle several difficulties: **1. Missing values in dynamic features**. Many features are unavailable at the beginning of petitions, which makes it challenging to model petitions at the early stage due to the paucity of available features. As shown in Figure 1, the input consists of eight features. However, only four features "text of petition content", "petitioner's profile","number of

[1]Change.org: https://www.change.org/.

[2]Stop contributing to massive food waste in the us-victory: https://www.change.org/p/whole-foods-and-walmart-stop-contributing-to-massive-food-waste-in-the-us-31

978-1-5386-9159-5/18/$31.00 ©2018 IEEE
DOI 10.1109/ICDM.2018.00069

signatures" and "petition comments" are accessible at the first day. One more feature "topic labels of the petitions" are available at the second day. **2. Strong uncertainty in petition prediction.** The victory of a petition is largely influenced by a lot of factors and thus is challenging and with high uncertainty. Thus, uncertainty estimation is an important aspect of petition victory prediction because it provides an accuracy estimation of the uncertainty of label predictions. However, evaluating the uncertainty of labels is difficult due to two reasons: on one hand, the uncertainty follows a temporal dependency: that is, the uncertainty is very high as a new petition is launched, then drops towards to zero as petition process approaches to the end since the progress becomes clearer and clearer, which is shown in Figure 1; and more importantly, all features are subject to change during the petition process while most current classifiers assume the fixed input. For example, in Figure 1, the feature "updated appeal process" which provides updated information about petition process up to now is dynamic. Due to unknown patterns of dynamic features, it is challenging to estimate label uncertainty by fitting a monotonically decrease function. **3. Unknown labels for ongoing petitions.** Most petition processes will persist for a long period of time due to a required large collection of signatures. Therefore, the label of a petition is unknown for most of the time. In Figure 1, the petition is ongoing for the first three days and is declared victory at the fourth day. In other words, the labels of this petition on the first three days are missing. In practice, some petitions run without labels for years. It will be a huge information loss if these petitions which provide abundant feature information are discarded. **4. Scalability regarding increasing features and petitions.** The task of petition victory prediction is conducted online and hence computational scalability is critical. However, this is challenging for several reasons, including (1) a fast growing number of online petitions; (2) an increasing number of features for each petition and (3) frequent updates of petition information. That means that even for a medium size dataset of 10,000 petitions, and even in the case where we consider only the information derived from comments of signers of a petition, we may have tens of thousands of signer comments per petition, each represented by a thousand numeric features. This yields a dataset having hundreds of billions of data points, which entails massive storage and time-consuming computation. As a result, petition victory prediction requires large-scalable optimization algorithms.

In order to simultaneously address all these technical problems, we propose a novel Multi-task Learning model with Uncertainty Estimation (MLUE). Specifically, given data with dynamic features and missing values, we divide data into several increasing feature blocks according to missing patterns. A multi-task learning strategy is presented to handle missing values. In order to deal with unlabeled labels and uncertainty estimation of petitions, we propose a novel measurement to evaluate uncertainty and assign labels to unlabeled petitions as accurately as possible. As for the challenge of scalability, we develop an EM-like optimization algorithm to process large-scale petition data in a decentralized fashion: in the E-step, we propose a dynamic programming method to estimate the labels of unlabeled petitions; in the M-step, we introduce an Alternating Direction Method of Multipliers (ADMM) [4] approach to optimize parameters in parallel. The parallel implementation of the MLUE solves the problem of large-scale petition processing.

The main contributions of our research are summarized as follows:

- **Design a MLUE framework to address the petition victory prediction problem.** A generic framework is proposed for petition victory prediction to deal with missing values in dynamic features. We give the definition of increasing feature blocks and present a multi-task learning model with chain structure to handle missing values. Our framework can be generalized into more window sizes.
- **Propose a novel criterion to estimate uncertainty.** To estimate the uncertainty of predicted petition labels, we propose a new criterion to compare the performance difference between two classifiers in two adjacent blocks. This criterion can also generalized to more window sizes.
- **Develop an efficient nonconvex optimization algorithm.** The optimization problem is non-convex due to the fact that decision variables involve discrete unknown labels. An effective EM-like algorithm is developed to optimize parameters efficiently and accurately, which is scalable to the number of petitions and features. A dynamic programming method and an ADMM method are presented in the E-step and the M-step, respectively.
- **Conduct extensive experiments for performance evaluations.** Experiments on six real petition datasets show that our MLUE dominated other models. Residuals and scalability are explored as well. In the sensitivity analysis, we analyze the sparsity patterns of our MLUE and investigate key parameters.

The rest of the paper is organized as follows. In Section II, we summarize recent research work related to this paper. In Section III, we present the problem formulation. In Section IV, we propose the novel MLUE framework. In Section V, we develop an effective EM-like optimization algorithm. In Section VI, extensive experiments are conducted to validate the effectiveness of our model. Section VII concludes by summarizing the whole paper.

II. Related Work

The idea of mining and predicting petitions obtained from OPPs has recently been proposed in a vision paper by Li et al. [14]. This vision paper broadly surveys the applications of using OPP for petition victory prediction, but does not give an solutions, algorithms or implementations to achieve this goal. To the best of our knowledge, no other existing research employs OPP data for victory prediction. Yet, there is a large research body related to the sub-problems that we encountered to build our petition victory prediction framework. These sub-problems include the estimation of missing values, uncertainty

estimation of unlabeled data and multi-task learning in time-series data. This research is surveyed in the following.

Estimation of missing values: The figure and analysis of missing values have been explored in some early literatures, surveyed in [10]. Other early work estimates missing values from observed values [8]. These methods performed well when missing values were rare, but they were ineffective when the proportion of missing values became large. To deal with such sparse data, Hernandez et al. proposed a probabilistic matrix factorization model for collaborative filtering to handle non-random missing values [11]. Yuan et al.[24] and Xiang et al. [22] utilized multi-task learning to learn a consistent feature selection pattern across different missing groups. Zhao et al. presented a multi-source learning framework to address hierarchical missing values [31]. Some deep learning literatures also deal with missing data: Che et al. developed a recurrent neural network model and represented missing patterns as masking and time interval [5]; Lipton et al. made use of binary indicators to directly model missing clinical time series data [17]. However, none of these approaches focuses specifically on missing values in dynamic features.

Uncertainty estimation of unlabeled data: Recently, the uncertainty estimation of unlabeled data has attracted the attention of researchers. Several ideas from semi-supervised learning and domain adaptation were also relevant[16][19]. Some work attempted to approximate the distribution of unlabeled data either by importance weighting[20] or mis-specified models[3]. See [21] for a review. A.P.Dawid and A.M.Skene estimated error-rates for polytomous facets when true responses from patients were unavailable in early days [6]. However, the problem of uncertain uncertainty estimation was first formally introduced by Donmez et al. [7]. Inspired by Dawid-Skene estimator, Zhang et al. proposed a two-stage EM-based algorithm for multi-class crowd labeling problem in the setting of a non-convex log-likelihood function [27]. Platanios et al. estimated the accuracy of unlabeled data by a collection of competing classifiers [18]. Other work such as Jaffe et al. considered dependent classifiers [13] and Balasubramanian et al. introduced a continuous loss function and showed that the distribution of errors was often approximately Gaussian [2]. However, none of these literatures focuses on a time-dependent uncertainty estimation.

Multi-task learning in time series data: Multi-task learning (MTL) aims to leverage useful information from multiple related tasks to improve the generalization of models. Typically, multi-task learning approaches can be categorized into many classes such as feature learning approach, low-rank approach and task clustering approach. For a completed list of categorization, see [26] for review. MTL is widely applied in time series data as well as other domains. For example, Zhao et al. proposed two MTL models by considering heterogeneous relationships for different location resolutions in the spatial event forecasting problem [29]; Xu et al. proposed an online MTL framework with temporal smoothness for ensemble forecasting [23]; Zhou et al. developed a MTL model to predict the disease progression with temporal regularization [33]; Zhao et al. utilized multi-lingual indicators to forecast social events [30]. However, to the best of our knowledge, ours is the first work to apply MTL for petition victory prediction.

TABLE I
IMPORTANT NOTATIONS AND DESCRIPTIONS

Notations	Descriptions
$Y_{i,t}$	The label of the i-th petition at time t.
$X_{i,t}$	The i-th input petition matrix on the date t.
d_i	the label time of the i-th petition.
$IFB(j)$	The j-th Increasing Feature Block.
$\mathscr{A}_{T_j}(X_{i,t})$	the set of available features of the data $X_{i,t} \subset IFB(j)$.
β_j	The coefficient feature vector of $IFB(j)$.
n	The number of petitions.
m	The number of features.
A	The number of increasing feature blocks.
T	All the time intervals.
T_j	The time interval of the j-th Increasing Feature Block.
λ_1, λ_2	regularization parameters.
w	window size.

III. PROBLEM SETUP

In this section, the problem addressed by this research is formulated. Section III-A defines the problem of petition victory prediction; and Section III-B discusses challenges of the problem.

A. Problem Formulation

We firstly introduce necessary mathematical notations displayed in Table I, and then provide the problem formulation.

A petition will be labeled as victorious if the required number of signatures is satisfied or the appeals of the petition-launcher have been addressed by the decision-makers within a limited time interval. Otherwise, the petition is labeled as failed. A petition is launched by the petitioner and then it goes through a period of time until reaching the time point when we know whether the it is labeled as a victory or not, called label time. Define T as all the time intervals. The label time of the i-th petition is defined as $d_i \in T$. Therefore, for all the time intervals before d_i, the i-th petition is "unlabeled" which means its victory (or failure) status is unknown, while the petition is "labeled" when after d_i, namely $t \geq d_i$ where $t \in T$ is current time. $X_i = \{X_{i,t}\}_{t \in T}$ denotes the set of the i-th petition in all the different time intervals T where $X_{i,t} \in \mathbb{R}^m$ denotes the feature vector of the i-th petition at time $t \in T$ and m is the number of features of a petition. X_i can be split into two subsets by the label time l_i: The unlabeled petition set is $X_i^u = \{X_{i,t}\}_{t<d_i}$ while the labeled set is $X_i^l = \{X_{i,t}\}_{t \geq d_i}$. $X^l = \{X_i^l\}_{1 \leq i \leq n}$ and $X^u = \{X_i^u\}_{1 \leq i \leq n}$ are denoted as the labeled petition set and the unlabeled petition set, respectively. The petition set is denoted as $X = X^l \cup X^u = \{X_i\}_{1 \leq i \leq n}$ where n is the number of petitions. For any $X_{i,t} \in X_i^l$, it has a corresponding label $Y_{i,t} \in \{-1, 1\}$ denotes the label of the i-th petition at time t: $Y_{i,t} = 1$ means that the i-th petition is under victory status at time t while $Y_{i,t} = -1$ implies that it is under failure status at time t. For any $X_{i,t} \in X_i^u$, its label $Y_{i,t}$ is unknown. $Y_i = \{Y_{i,t}\}^{t \in T}$ denoting the label set of the i-th petition. $Y = \{Y_i\}_{1 \leq i \leq n}$ denotes the labels of all petitions.

The petition victory prediction problem can be formulated as follows:

Problem Formulation: Given the petition vector $X^u_{i,t}$ with label $Y_{i,t}$, the goal of the problem is to predict whether the i-th petition will succeed at time $t+\tau$ by learning the mapping $f : X_{i,t} \rightarrow Y_{i,t+\tau}$, where τ is the lead time.

B. Challenges

In order to solve the petition victory prediction problem, we still need to tackle several challenges: 1). The available feature set of a petition in the beginning is very small, while more features are gradually involved along with time. In other words, for the i-th petition, the available feature set of $X_{i,t}$ at time t is smaller than that of $X_{i,s}$ at time s where $t < s$. 2). According to the problem formulation, for any $X_{i,t} \in X^u$, its label is unknown. If we discard them, a majority of information is at a loss because the number of unlabeled petition samples $|X^u|$ outnumber greatly that of labeled petition samples $|X^l|$. 3). Even we predict the labels from the set of unlabeled petitions X^u, the uncertainty of these predictions is unknown. Wrong predictions may introduce extra noise to the classifier. 4). An optimization algorithm to solve this problem meets the difficulty of a growing amount of computation as the petition set $|X|$, the time intervals T, and the number of features increase jointly. Thus in the next two sections, a novel multi-task learning model is proposed to address these problems in turn.

IV. MLUE Model

In this section, we develop the novel MLUE model. Specifically, we introduce a new definition to handle missing values in Section IV-A; Section IV-B provides a criterion to discuss uncertainty estimation problem; Section IV-C gives the complete framework of our model; In Section IV-D, we generalize our model to more window sizes.

A. Missing Values in Dynamic Features

One simple method to deal with missing values is to impute them with zeros. However, it fails to consider temporal smoothness of time series data. To deal with this problem, we consider multi-task learning as a good strategy to deal with missing values [9]: it divides data into several building blocks and learn their own parameters by relatedness among building blocks, which are defined on time intervals. We introduce the definition of increasing feature blocks as follows.

Definition 1 (Increasing Feature Block)**.** *The j-th **Increasing Feature Block (IFB)** $IFB(j)$ is a block of petition sets $\{X_{i,t}\}^{t\in T_j}_{1\le i\le n}$ ($T_j \subset T$) that share the same available feature sets generated by interval T_j. Define $\mathscr{A}_{T_j}(X_{i,t})$ as the set of available features of the data $X_{i,t} \subset IFB(j)$. Assume the total number of IFBs is A, then they must satisfy the following three criteria:*

- *Completeness:* $T = \cup_{j=1}^{A} T_j$.
- *Coherence:* $\forall p, q \in T_j : \mathscr{A}_{T_j}(X_{i,p}) = \mathscr{A}_{T_j}(X_{i,q})$.
- *Orderliness:* $\forall q \in T_a, p \in T_b, a < b : \mathscr{A}_{T_a}(X_{i,q}) \subsetneqq \mathscr{A}_{T_b}(X_{i,p})$.

According to the orderliness properties of IFB, each feature block has an increasing set of features. Suppose β_j and b_j are denoted as the weight and the intercept learned by the $IFB(j)$, respectively, then the weight and intercept over all IFBs are denoted as $\beta = \cup_{j=1}^{A}[\beta_j]$ and $b = \cup_{j=1}^{A}[b_j]$, respectively.

B. Uncertainty Estimation

In practice, a petition may take years to succeed, which means the label time d_i could be very large. As a result, most of the i-th petition data points whose time is earlier than d_i are unlabeled. The majority of unlabeled petitions provide useful information from many available dynamic features. Therefore, discarding them will lead to a huge information loss. In other words, we can achieve better prediction performance with the introduction of available features from unlabeled petitions. However, as shown in Figure 1, in the beginning of a petition, only few features are available which makes the petition-victory prediction highly uncertain. The uncertainty in prediction would decrease as the number of features increases. And typically the prediction is more and more certain towards the label, because the increase of the number of features more and more clearly indicates the victory or failure of the petition.

Therefore, given the label time d_i, we extend the idea of [16] and design a novel criterion to characterize the uncertainty in prediction by comparing the predicted labels of the i-th petition in $IFB(j-1)$ and $IFB(j)$, respectively. First, if the prediction in $IFB(j-1)$ is inaccurate while that in $IFB(j)$ is accurate, more certainty is obtained and thus we earn certainty along with time. Mathematically, this is denoted by $earn(Y_{i,q}, Y_{i,p}, Y_{i,d_i})$, namely, the accuracy earning of the classifier: $Y_{i,p} = Y_{i,d_i}$ but $Y_{i,q} \neq Y_{i,d_i}$ where $p \in T_j$ and $q \in T_{j-1}$. In contrast, if the prediction in $IFB(j)$ is not accurate while that in $IFB(j-1)$ is accurate, less certainty is earned and hence we lose certainty across time. Likewise, we denote $lose(Y_{i,q}, Y_{i,p}, Y_{i,d_i})$ as the accuracy losing of classifier, namely $Y_{i,p} \neq Y_{i,d_i}$ but $Y_{i,q} = Y_{i,d_i}$ where $p \in T_j$ and $q \in T_{j-1}$. More specifically:

$$\begin{aligned} earn(Y_{i,q}, Y_{i,p}, Y_{i,d_i}) &= I(Y_{i,q} \neq Y_{i,d_i}) I(Y_{i,p} = Y_{i,d_i}) \\ &= (1 - Y_{i,q}Y_{i,d_i})(1 + Y_{i,p}Y_{i,d_i})/4 \\ lose(Y_{i,q}, Y_{i,p}, Y_{i,d_i}) &= I(Y_{i,q} = Y_{i,d_i}) I(Y_{i,p} \neq Y_{i,d_i}) \\ &= (1 + Y_{i,q}Y_{i,d_i})(1 - Y_{i,p}Y_{i,d_i})/4 \end{aligned}$$

where $I(\bullet)$ is an indicator function.

In all, since we will always want to maximize the prediction accuracy, and thus we need to minimize the lose and maximize the earn, which is equal to miminize the following:

$$R(Y_{i,q},Y_{i,p},Y_{i,d_i}) = lose(Y_{i,q},Y_{i,p},Y_{i,d_i}) - earn(Y_{i,q},Y_{i,p},Y_{i,d_i})$$

Obviously, more value of $lose(Y_{i,q}, Y_{i,p}, Y_{i,d_i})$ and less value of $earn(Y_{i,q}, Y_{i,p}, Y_{i,d_i})$ lead to more uncertainty $R(Y_{i,q}, Y_{i,p}, Y_{i,d_i})$. The uncertainty function of the i-th petition $R(Y_i)$ is the sum of the uncertainty function $R(Y_{i,q}, Y_{i,p}, Y_{i,d_i})$ at IFB-adjacent data points p and q: $R(Y_i) = \sum_{j=2}^{A} \sum_{p\in T_j} \sum_{q\in T_{j-1}} R(Y_{i,q}, Y_{i,p}, Y_{d_i})$. The overall uncertainty function $R(Y)$ is the sum of uncertainty

functions of all petitions: $R(Y) = \sum_{i=1}^{n} R(Y_i)$.

With the overall uncertainty function $R(Y)$, we encourage accurate labels are assigned at the early days and the weight β_j and intercept b_j in the $IFB(j)$ can be learned by accurately assigned labels with limited available features.

C. Overall Model

The above consideration of IFB and uncertainty estimation lead to a new mutli-task learning framework that jointly minimizes the empirical error and the uncertainty:

$$(Y, \beta^*, b^*) = \arg\min_{Y,\beta,b} Loss(Y; \beta, b) + \lambda_1 \Omega(\beta) + \lambda_2 R(Y)$$
$$s.t. \forall q \leq p, Y_{i,q} \leq Y_{i,p} \quad (1)$$

where $Loss(Y; \beta, b) = 1/|X| \sum_{j=1}^{A} \sum_{X_{i,t} \in IFB(j)} \max(0, 1 - Y_{i,t}(X_{i,t}^T \beta_j + b_j))$ is a hinge loss function, $\Omega(\beta) = \sum_{j=1}^{A-1} \|\beta_j, \beta_{j+1}\|_{2,1}$ is a chain-structure penalty term which enforces two IFB-adjacent petition data points to share the same sparsity pattern, and $\lambda_1, \lambda_2 > 0$ are regularization parameters to control the balance between the loss function $Loss(Y; \beta, b)$, the chain-structure penalty $\Omega(\beta)$ and the uncertainty function $R(Y)$. The inequality constraint $\forall q \leq p, Y_{i,q} \leq Y_{i,p}$ means that a petition label at a given time is subject to its successors, i.e. if $Y_{i,p} = -1$, then $Y_{i,q} = -1$ for $0 \leq q \leq p$.

D. Model Generalization

Suppose the window size w is defined as the number of adjacent IFBs to consider, then we consider our model with $w = 2$ in the previous sections. Our model can be further generalized to more window sizes, i.e. the predicted labels in $IFB(j)$ and $IFB(j+1), \cdots, IFB(j+w-1)$. The novelty of this generalization consists in the exponential weighted smoothing technique: the more adjacent two IFBs are, the more weight we assign. The uncertainty function of the i-th petition $R(Y_i)$ is defined recursively as follows:

$$R^{(1)}(Y_i) = \sum_{j=w}^{A} \sum_{p \in T_j} \sum_{q \in T_{j-w+1}} R(Y_{i,q}, Y_{i,p}, Y_{d_i})$$
$$R^{(k)}(Y_i) = \alpha \sum_{j=w-k+1}^{A} \sum_{p \in T_j} \sum_{q \in T_{j-w+k}} R(Y_{i,q}, Y_{i,p}, Y_{d_i})$$
$$+ (1-\alpha) R^{(k-1)}(Y_i), \ (2 \leq k \leq w-1)$$

where $R^{(k)}(Y_i)$ is the $k-th$ iteration of computing $R(Y_i)$, $R(Y_i) = R^{(w-1)}(Y_i)$ and $0 \leq \alpha \leq 1$ is a weight parameter. Correspondingly, the chain-structure penalty with windows size w is denoted as:

$$\Omega(\beta) = \sum_{j=1}^{A-w+1} \|\beta_j, \cdots, \beta_{j+w-1}\|_{2,1}.$$

V. Optimization Algorithm

The Equation (1) is non-convex due to the fact that decision variables involve discrete variables Y. An EM-like algorithm is proposed to minimize Y and β alternatively. The overall algorithm is presented in Algorithm 1. The E-step and M-step in Line 3 and 4 update Y^{k+1} and (β^{k+1}, b^{k+1}) alternately. Therefore, the following two sections focus on two subproblems, respectively. Specifically, Section V-A discusses a dynamic programming approach to update Y; in the Section V-B, an Alternating Direction Method of Multipliers (ADMM) [4] algorithm is developed to update β and b.

Algorithm 1 Overall Algorithm

Require: X, Y, λ_1, λ_2.
Ensure: Y, β, b
1: Initialize β, b, Y, $k = 0$.
2: **repeat**
3: E-step: update Y^{k+1} in Equation (2) while fixing β and b.
4: M-step: update (β^{k+1}, b^{k+1}) in Equation (3) while fixing Y.
5: $k = k + 1$.
6: **until** convergence.
7: Output Y, β, b.

A. E-step: Assigning Labels to Petitions

The E-step update is formulated as follows:

$$Y^{k+1} \leftarrow \arg\min_{Y} Loss(Y; \beta, b) + \lambda_2 R(Y)$$
$$s.t. \forall p \leq q, Y_{i,p} \leq Y_{i,q} \quad (2)$$

Given the weight β and the intercept b, solving Equation (2) amounts to label assignments to unlabeled petitions. We design a dynamic programming approach to solve this subproblem described in Algorithm 2. The label set of the i-th petition Y_i is updated independently of other petitions. The time complexity of this algorithm is $O(n^2 A)$.

One important aspect of this algorithm is the initial value of β and b. This is because the output of β and b is sensitive to initialization. It is recommended that an initial point of β and b for each task be set to of a classifier trained independently.

Algorithm 2 E-step: Assign Labels to Unlabeled Petitions.

Require: λ_2, β, b,X
Ensure: Y
1: **for** i=1 to n # The outer loop can be implemented in parallel. **do**
2: **for** j=A-1 to 1 **do**
3: Find label set $\{Y_{i,t}\}$ $s.t.$ $X_{i,t} \in IFB(j+1)$.
4: **if** $\exists t$ $s.t.$ $Y_{i,t} = -1$ **then**
5: **for all** $X_{i,t} \in IFB(j)$ **do**
6: $Y_{i,t} = -1$
7: **end for**
8: **else**
9: **for all** $X_{i,t} \in IFB(j)$ **do**
10: $Y_{i,t} = \arg\min_{Y_{i,t}} \max(0, 1 - Y_{i,t}(X_{i,t}^T \beta_j + b_j)) + \lambda_2 \sum_{p \in T_{j+1}} R(Y_{i,t}, Y_{i,p}, Y_{d_i})$
11: **end for**
12: **end if**
13: **end for**
14: **end for**
15: Output Y.

B. M-step: Solving the Chain-structure Multi-task Learning Problem

The M-step update is formulated as follows:

$$(\beta^{k+1}, b^{k+1}) \leftarrow \arg\min_{\beta,b} Loss(Y; \beta, b) + \lambda_1 \Omega(\beta) \quad (3)$$

The chain-structure penalty $\Omega(\beta)$ makes this subproblem difficult to solve because most β_js appear twice in ℓ_{21} penalties. We develop an ADMM-based algorithm to handle it. By

introducing two auxiliary variables S and ϕ, Equation (3) can be transformed equivalently to the following formulation:

$$\min_{\beta,b,S} Loss(Y;S) + \lambda_1\Omega(\beta,\phi)$$
$$s.t.\ \beta_{j+1} = \phi_j;\ S_{i,t} = X_{i,t}^T\beta_j + b_j, X_{i,t} \in IFB(j). \quad (4)$$

where $Loss(Y;S) = (1/|X|)\sum_{j=1}^{A}\sum_{X_{i,t}\in IFB(j)} \max(0, 1-Y_{i,t}S_{i,t})$, $\Omega(\beta,\phi) = \sum_{j=1}^{A-1}\|\beta_j,\phi_j\|_{21}$ and $\phi = \cup_j^{A-1}[\phi_j]$.

The augmented Lagrangian function of Equation (4) is:

$$L_\rho(\phi,\beta,b,S,u,v) = Loss(Y;S) + \lambda_1\Omega(\beta,\phi) +$$
$$(\rho/2)\|S - X\beta - b + u\|_2^2 + (\rho/2)\sum_{j=1}^{A-1}\|\beta_{j+1} - \phi_j + v_j\|_2^2$$

where $\rho > 0$ is a penalty parameter and u and v are dual variables. The M-step update is shown in Algorithm 3. Concretely, Lines 10-14 calculate residuals and Lines 3-9 update each parameter alternately by solving the subproblems described below.

Algorithm 3 M-step: Chain-structure Multi-task Learning

Require: X, Y, λ_1.
Ensure: β, b
1: Initialize S, β, b, $\rho = 1$, $k = 0$.
2: **repeat**
3: Update ρ^{k+1} if necessary.
4: Update S^{k+1} by Equation (5).
5: **for** j between 1 and $A-1$ **do**
6: Update $(\beta_j^{k+1}, b_j^{k+1}, \phi_j^{k+1})$ by Equation (6).
7: **end for**
8: Update $(\beta_A^{k+1}, b_A^{k+1})$ by Equation (7).
9: Update (u^{k+1}, v^{k+1}) by Equation (8).
10: **for** j between 1 and $A-1$ **do**
11: $r_j = \beta_{j+1} - \phi_j$
12: **end for**
13: $r_A = S^{k+1} - X\beta^{k+1} - b^{k+1}$.
14: $r = \sqrt{\sum_{j=1}^{A} r_j^2}$.#Calculate residual.
15: $k = k + 1$.
16: **until** convergence.
17: Output β, b.

1. Update S^{k+1}.
The auxiliary variable S is updated as follows:

$$S^{k+1} \leftarrow \arg\min_S Loss(Y;S) + \rho^{k+1}/2\|S - X\beta^k - b^k + u^k\|_2^2 \quad (5)$$

This subproblem has a closed solution as follows:

$$S_{i,t}^{k+1} = Z_{i,t} + Y_{i,t}/(|X|\rho^{k+1}) * \max(sgn(1 - Y_{i,t}Z_{i,t}), 0)$$

where $Z_{i,t} = X_{i,t}^T\beta_j^k + b_j^k - u_{i,t}^k$.

2. Update $(\beta_j^{k+1}, b_j^{k+1}, \phi_j^{k+1})$,$(1 \le j \le A-1)$.
The variable triple (β_j, b_j, ϕ_j) is updated as follows:

$$(\beta_j^{k+1}, b_j^{k+1}, \phi_j^{k+1}) \leftarrow \arg\min_{\beta_j,b_j,\phi_j} \lambda_1\|\beta_j,\phi_j\|_{2,1}$$
$$+ (\rho^{k+1}/2)\sum_{X_{i,t}\in IFB(j)}^{T}(S_{i,t}^{k+1} - X_{i,t}\beta_j - b_j + u_{i,t}^k)^2$$
$$+ (\rho^{k+1}/2)\|\beta_{j+1}^k - \phi_j + v_j\|_2^2 \quad (6)$$

This subproblem is a least square loss function with a ℓ_{21} penalty. An accelerated gradient descent method [32] is applied to solve this problem.

3. Update $(\beta_A^{k+1}, b_A^{k+1})$.
The variable pair (β_A, b_A) is updated as follows:

$$(\beta_A^{k+1}, b_A^{k+1}) \leftarrow \arg\min_{\beta_A,b_A}(\rho^{k+1}/2)\|\beta_A - \phi_{A-1}^{k+1} + v_A^k\|_2^2$$
$$+(\rho^{k+1}/2)\sum_{X_{i,t}\in IFB(A)}^{T}(S_{i,t}^{k+1} - X_{i,t}\beta_A - b_A + u_{i,t}^k)^2 \quad (7)$$

TABLE II
FEATURE REPRESENTATION OF ALL PETITIONS

Petition Field	Number of Features
basic properties	10
petition topic	21
petition tag	128
petition title	288
petition description	711
petition body	210
victory description	79
petition comment	74
all	1521

TABLE III
LABEL INFORMATION OF SIX DATASETS.

Country	Victorious #Petitions	Failed #Petitions	Ongoing #Petitions
Philippines	60	202	750
India	237	2,527	3,110
Germany	776	2,691	5,594
Australia	479	1,374	2,525
Canada	398	1,475	1,951
United States	4,081	7,405	18,404

This subproblem is a sum of two least sqaure error function and hence has a closed-form solution.

4. Update (u^{k+1}, v^{k+1})
The dual variable pair (u, v) is updated as follows:

$$u^{k+1} \leftarrow u^k + S^{k+1} - X\beta^{k+1} - b^{k+1}.$$
$$v_j^{k+1} \leftarrow v_j^{k+1} + \beta_j^{k+1} - \phi_{j-1}^{k+1}. \quad (8)$$

VI. EXPERIMENT

In this section, we evaluate the MLUE using six real online petition datasets[3]. The effectiveness and the efficiency of the MLUE are assessed against several state-of-the-art methods. They were conducted on a 64-bit machine with Intel(R) core(TM)processor (i7-6820HQ CPU@ 2.70GHZ) and 16.0GB memory.

A. Experiment Setup

1) Input Petition Data Retrieval: The petitions description data in this paper were retrieved by the following process: First, we queried the Change.org[4] API to obtain information from 54,039 petitions during Jan 1, 2009 and Dec 17, 2017 from six countries: Philippines, India, Germany, Australia, Canada, and United States. All experiments were analyzed in compliance with the Change.org policies[4]. Second, all corresponding comments were retrieved by Change.org API again. The feature representation of each petition is illustrated in Table II. All fields except basic petition properties were unstructured raw texts and were represented by a set of keywords provided by domain experts. Basic petition properties such as "calculated goal" and "weekly signature count" were obtained originally on the Change.org website. The label information of six datasets, which was extracted by the "is victory" field, is detailed in Table III.

[3]The data and the code are available at http://mason.gmu.edu/~lzhao9/
[4]Change.org: https://www.change.org/policies

2) Metrics and Parameter Settings: In the experiments, five metrics were utilized to evaluate model performance: the Accuracy (ACC) is the ratio of accurately labeled petitions to all petitions; the Precision (PR) is the ratio of accurately labeled as positive petitions to all labeled as positive petitions; the Recall (RE) defines the ratio of accurately labeled as positive petitions to all positive petitions; the F-score (FS) is the harmonic mean of precision and recall; the Receiver Operating Characteristic (ROC) curve delineates the classification ability of a model as its discrimination threshold varies; the Area Under ROC (AUC) is an important measurement of classification ability.

The lead time τ defined in the problem formulation was set to one year. The number of IFBs A was set to 3. Two tuning variable λ_1 and λ_2 were set to 0.01 and 0.01 based on the AUC of the 5-fold cross-validation on the training set, respectively. We applied the under-sampling strategy [25] to avoid the class imbalance problem shown in Table III. The penalty parameter of the ADMM ρ was set to 1. The maximum number of iteration in the EM-like algorithm was set to 20. The maximum number of iteration in the ADMM algorithm was set to 100.

3) Comparison Methods: Since our method utilized multi-task learning strategies and unlabeled data, two types of state-of-the-art methods: multi-task learning and semi-supervised learning methods served as baselines for the performance comparison: the first three were multi-task classification approaches while the last three were semi-supervised classification methods. For the multi-task classification approaches, all tasks were grouped by the missing patterns. For the semi-supervised classification methods, the missing values were filled with 0. All parameters were set by 5-fold cross-validation on the training set.

1. Constrained Multi-Task Feature Learning I (cMTFL-I) [28]. cMTFL-I aims to directly control the number of features to be selected rather than control the group sparsity.

2. convex relaxed Clustered Multi-Task Learning(CMTL) [32]. CMTL assumes that tasks in the same group are more similar to the tasks in the different group. The loss function was set to the logistic log function.

3. multi-task learning with Joint Feature Selection (JFS) [1], [32]. JFS is one of the most commonly used strategies in multi-task learning. It captures the relatedness of multiple task by a constraint of weight matrix to share a common set of features, which can be realized by a ℓ_1/ℓ_2 regularization to ensure group sparsity. The loss function was set to the logistic log function.

4. WEakly LabeLed Support Vector Machines (WELLSVM) [15]. WELLSVM is proposed to solve the problem of weakly labeled data by a label generation strategy. As a convex relaxation of Mixed-Integer Programming (MIP) problem, WELLSVM can be solved by a sequence of subproblems to ensure scalability.

5. Stochastic optimization for Cluster Kernel(SoCK) [12]. SoCk is a graph-based semi-supervised learning method subject to the storage budget limit. It finds the low-rank approximation of the similarity matrix. SoCK is optimized stochastically and converges globally to the low-rank approximator.

6. Nystrom Cluster Kernel (NysCK) [12]. NysCK is very similar to the SoCK except that the NysCK is optimized by the Nystorm method and is suitable for online learning.

B. Performance

In this section, experimental results for the MLUE are analyzed for all the comparison methods. Table IV summarizes prediction results of the MLUE compared with other methods on the six petition datasets.

1) Model Performance on Six Petition Datasets: The first part of the results demonstrated in Table IV indicated that the MLUE outperformed any other baseline in any petition dataset. It ranked the first in the four metrics out of five except PR and RE. As the most important metric of five, the AUC of the MLUE dominated all baselines: the AUC of the MLUE was higher than 0.82 in six petition dataset and performed the best in the United States petition dataset, which achieved as high as 0.914, while all four baselines were lower than 0.8 in the AUC. When it came to the ACC, the MLUE was again superior to others: all of ACCs reached over 0.8 except the Philippines dataset. However, none of comparison methods had a performance of over 0.8 in the ACC. Due to the advantage over others in the ACC, the MLUE also performed competitively in the PR and FS. For example, its performance was 0.887 and 0.8110 on the PR and FS metrics in the Australia dataset, respectively, whereas these of JFS were 0.4582 and 0.6278, respectively. The MLUE also achieved a competitive score in the RE metric, surpassing 0.8 in the Germany dataset, whereas the RE of the WELLSVM was only around 0.66. The superiority of the MLUE consisted in effective utilization of unlabeled petitions as well as sparsity perseverance between two adjacent IFBs, while multi-task learning methods lacked the ability to leverage unlabeled petitions and the three semi-supervised learning methods suffered from missing values. As for competition among six baselines, the WELLSVM outperformed multi-task learning methods thoroughly. The ACC of the WELLSVM was higher than 0.67 in the Canada dataset, whereas the best score of multi-task learning methods, which was achieved by the cMTFL-I, was only lower than 0.52. The AUC of the WELLSVM was in the vicinity of 0.78 in the Australia dataset. However, none of three multi-task learning methods scored over 0.7. Interestingly, we found that the JFS achieved the best of the RE metric in every petition dataset. As for the SoCK and NysCK, they achieved higher performance than WELLSVM.

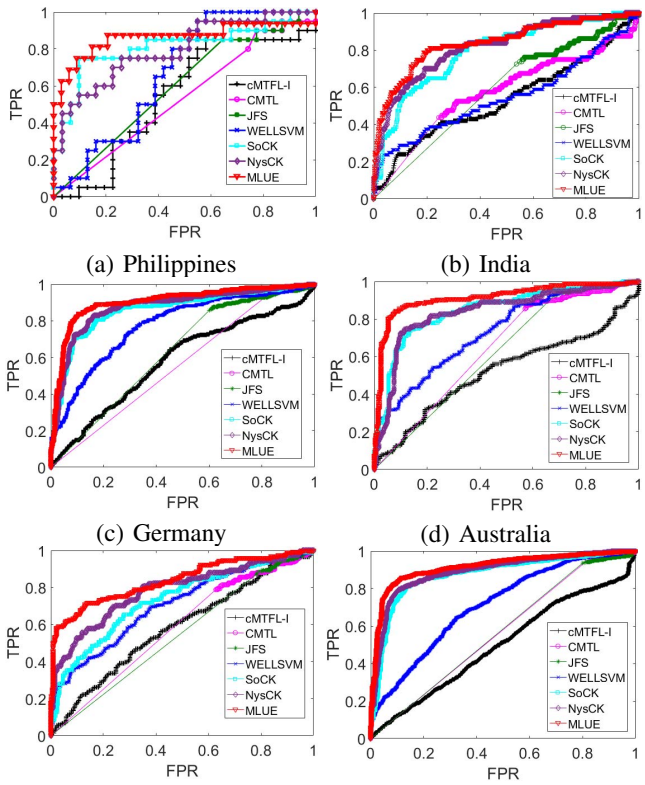
Figure 2 shows the ROC curves of the MLUE and baselines on the six petition datasets. In each ROC curve, the X axis and the Y axis denote False Positive Rate (FPR) and True Positive Rate (TPR), respectively. Overall, the ROC curve of the MLUE almost covered baselines, which was consistent with Table IV. The WELLSVM contained other three multi-task learning

methods. The SoCK and the NysCK performed the best of all baselines, which was a littel inferior to the MLUE. The cMTFL-I annd JFS were comparable to each other: they were both slightly better than the random guess.

TABLE IV
MODEL PERFORMANCE ON THE SIX PETITION DATASETS UNDER FIVE METRICS: MLUE OUTPERFORMED OTHERS ON FOUR METRICS OUT OF FIVE.

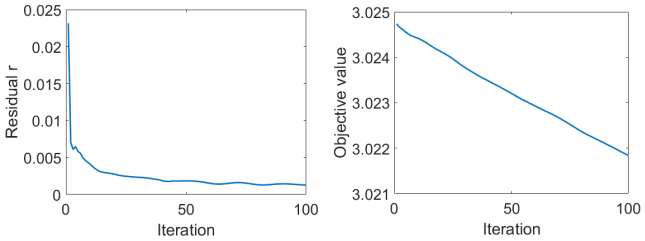
Philippines					
Method	ACC	PR	RE	FS	AUC
cMTFL-I	0.5869	0.4677	0.6716	0.5464	0.6113
CMTL	0.4014	0.3914	**1**	0.5618	0.5666
JFS	0.4726	0.4199	0.9609	0.5836	0.6330
WELLSVM	0.6301	0.8000	0.0616	0.1132	0.6669
SoCK	0.7777	0.8203	0.5241	0.6202	0.7843
NysCK	0.7678	0.8296	0.5232	0.6315	0.7667
MLUE	**0.7875**	**0.8777**	0.5300	**0.6591**	**0.8281**
India					
Method	ACC	PR	RE	FS	AUC
cMTFL-I	0.4616	0.1896	0.6429	0.2906	0.5638
CMTL	0.2680	0.1747	0.8895	0.2918	0.5988
JFS	0.2349	0.1775	**0.9537**	0.2982	0.5776
WELLSVM	0.4523	0.4268	0.6299	0.2732	0.5920
SoCK	0.8582	0.7622	0.2428	0.3664	0.7680
NysCK	0.8657	0.7334	0.3254	0.4506	0.8036
MLUE	**0.8780**	**0.7672**	0.4048	**0.5295**	**0.8271**
Germany					
Method	ACC	PR	RE	FS	AUC
CMTL	0.4385	0.4376	0.9976	0.6081	0.6107
rMTFL	0.4925	0.4370	0.5636	0.4855	0.4952
JFS	0.4400	0.4382	**0.9961**	0.6084	0.6016
WELLSVM	0.5318	0.6188	0.6630	0.4903	0.7015
SoCK	0.8083	0.7982	0.7519	0.7741	0.8737
NysCK	0.8357	0.8149	0.8076	0.8110	0.8900
MLUE	**0.8744**	**0.8814**	0.8239	**0.8514**	**0.9098**
Australia					
Method	ACC	PR	RE	FS	AUC
cMTFL-I	0.4086	0.4178	0.7709	0.5416	0.4829
CMTL	0.4580	0.4531	0.9595	0.6153	0.6130
JFS	0.4624	0.4582	**0.9991**	0.6278	0.6656
WELLSVM	0.6726	0.8284	0.4086	0.5144	0.7766
SoCK	0.7965	0.7897	0.7606	0.7729	0.8526
NysCK	0.8145	0.8163	0.7692	0.7905	0.8592
MLUE	**0.8412**	**0.8870**	0.7484	**0.8110**	**0.8915**
Canada					
Method	ACC	PR	RE	FS	AUC
cMTFL-I	0.5164	0.4162	0.6880	0.5154	0.5736
CMTL	0.3803	0.3754	0.9909	0.5445	0.6050
JFS	0.3838	0.3763	**0.9857**	0.5446	0.5625
WELLSVM	0.6707	**0.9551**	0.1287	0.2230	0.7345
SoCK	0.7397	0.7296	0.4838	0.5809	0.7733
NysCK	0.7774	0.7770	0.5677	0.6557	0.8058
MLUE	**0.8236**	0.8329	0.6614	**0.7365**	**0.8498**
United States					
Method	ACC	PR	RE	FS	AUC
cMTFL-I	0.5435	0.5588	0.9377	0.7003	0.4869
CMTL	0.5674	0.5695	**0.9799**	0.7203	0.5842
JFS	0.5735	0.5738	0.9773	0.7228	0.5996
WELLSVM	0.5359	0.7648	0.4915	0.4451	0.7250
SoCK	0.8202	0.8644	0.8112	0.8369	0.8719
NysCK	0.8347	0.8882	0.8117	0.8482	0.8880
MLUE	**0.8602**	**0.9247**	0.8212	**0.8698**	**0.9140**

2) The Effect of Iteration on the Residual and Objective Value: This part of experiments examined the effect of iterations on the residual and the objective value using the Philippines petition dataset. Figure 3(a) shows the change of residuals r with respect to iteration in the ADMM algorithm. The residual r began with a sharp decline and remained a smooth decline toward 0. This implies that tens of iterations were enough for practical optimization. The objective value of all training data and test data with regard to iteration is displayed in Figure 3(b). The objective value dropped linearly as expected.



(a) Philippines (b) India (c) Germany (d) Australia (e) Canada (f) United States

Fig. 2. ROC curves on six petition datasets: the MLUE was superior to baselines.



(a) Residual versus iteration (b) Objective versus iteration

Fig. 3. The effect of iteration on the residual and objective value: r and the objective value declined with iteration.

3) The Effect of λ_1 on the Sparsity Pattern: Because λ_1 controls the chain-structure penalty $\Omega(\beta)$, we investigated how λ_1 affected sparsity patterns. We chose three possible values of λ_1: 0.01, 0.1 and 1 while keeping $\lambda_2 = 0.01$. We denote $\#\beta(1)$, $\#\beta(2)$ and $\#\beta(3)$ as the number of features such that $(1)(\beta_1 = 0, \beta_2 = 0, \beta_3 \neq 0)$, $(2)(\beta_1 \neq 0, \beta_2 = 0, \beta_3 = 0)$ and $(3)(\beta_1 = 0, \beta_2 = 0, \beta_3 = 0)$, respectively. Table V shows the change of sparsity patterns decided by λ_1. Overall, sparsity patterns were more obvious as λ_1 increased. Concretely, the $\#\beta(1)$ and $\#\beta(3)$ were correlated positively with λ_1.

Furthermore, there was a boom of $\#\beta(1)$ as λ_1 ranged from 0.01 to 0.1. For example, when λ_1 varied from 0.01 to 0.1, $\#\beta(1)$ and $\#\beta(3)$ increased from 93 and 126 to 1,340 and 148 in the Canada dataset, respectively. Surprisingly, $\#\beta(2)$ was much fewer than $\#\beta(1)$ and $\#\beta(3)$ and reduced to 0 with the increase of λ_1. As an instance, $\#\beta(2)$ dropped from 32 to 0 as λ_1 varied from 0.01 to 1.

TABLE V
THE EFFECT OF λ_1 ON THE SPARSITY PATTERNS: THE SPARSITY PATTERNS WERE MORE OBVIOUS AS λ_1 INCREASED.

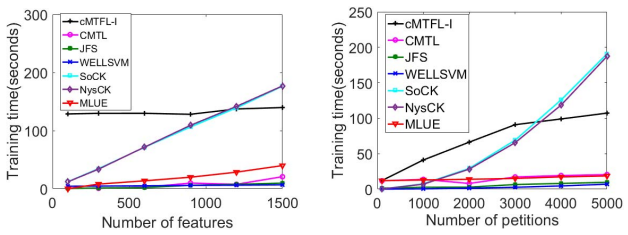
λ_1	Philippines			India			Germany		
	$\#\beta(1)$	$\#\beta(2)$	$\#\beta(3)$	$\#\beta(1)$	$\#\beta(2)$	$\#\beta(3)$	$\#\beta(1)$	$\#\beta(2)$	$\#\beta(3)$
0.01	61	32	375	230	8	204	234	11	98
0.1	1063	1	441	1272	1	218	1317	2	116
1	1079	0	443	1303	0	219	1401	0	118
λ_1	Australia			Canada			United States		
	$\#\beta(1)$	$\#\beta(2)$	$\#\beta(3)$	$\#\beta(1)$	$\#\beta(2)$	$\#\beta(3)$	$\#\beta(1)$	$\#\beta(2)$	$\#\beta(3)$
0.01	271	6	123	93	5	126	82	2	28
0.1	1357	0	137	1340	1	148	1231	0	30
1	1385	0	137	1373	0	149	1479	0	30

4) Scalability analysis: To examine the scalability of the MLUE, we measured the training times of all methods when varying number of features and number of petitions using 5,000 petitions in the United States petition dataset. The training time was calculated by the average of running 20 times.

Figure 4(a) compares the running time for all methods when the number of features varied from 100 to 1,500. Basically, the running time of all methods increased linearly with number of features. Among them, the WELLSVM and the JFS were implemented the most efficiently compared with other methods. The CMTL and the MLUE were also very efficient and consumed less than 50 seconds with 1,500 available features. The efficiency of the MLUE consisted in the parallel computing strategy of the ADMM as well as proper initialization of parameters. Obviously, the cMTFL-I was the slower than most baselines and kept steady no matter how many features were given. It tripled the training time of the MLUE when $1,500$ features were used for training. For the SoCK and the NysCK, they consumed similar running time and were the slowest among all baselines.

To examine the scalability for an increasing number of petitions, Figure 4(b) illustrates the running time of all methods when number of petition ranged from 100 to 5,000. Similar to the patterns shown in Figure 4(a), the running time of all methods increased linearly with number of petitions. This illustrated that our MLUE was scalable with respect to features and petitions. cMTFL-I was still the most time-consuming algorithm of all. However, the increased trend was suppressed with the introduction of more petitions.

5) Feature Analysis: In addition to petition victory prediction, the proposed MLUE can also find important features, which helps petition-launcher increase the possibility to set up victorious petitions. Table VI illustrates the top-10 features with the highest value of weight. The significance of features is ranked decreasingly with regard to weight value. Most



(a) Scalability on number of features

(b) Scalability on number of petitions

Fig. 4. Scalability on number of features and petitions: all methods increased linearly with the number of features and petitions.

features originate from the fields of petition tag and petition title. It seems that some types of petitions are more likely to succeed in different countries: For example, the decision-makers in India attach importance to petitions concerning food, environment, veterans and cancer; the petitions about disability rights, criminal justice and business and brands are prioritized in Germany; the petitions regarding healthcare, gay rights, disability rights and education are given priority by decision-makers in the United States. Thus the favorite petitions from the developing countries tend to focus on basic needs of the people while for developed countries education and human rights gain more concerns.

VII. CONCLUSION

Petition victory prediction is an important task in order to accelerate the process of petition reaction to public concerns. In this paper, we proposed a novel Multi-task Learning framework with uncertainty Estimation (MLUE) to predict potentially victorious petitions. Specifically, we divided data into different Increasing Feature Blocks (IFBs) according to missing patterns. We also presented a novel criterion to estimate uncertainty and introduced a chain-structured regularization term. Besides, we provided an EM-like algorithm to optimize the non-convex objective function in two stages: we proposed a dynamic programming method to make out the label of the unlabeled petitions in the E-step and introduced an ADMM-based method to optimize the coefficient of the classifier in the M-step. Various experiments on six petition datasets demonstrate that our MLUE dominated other baselines.

REFERENCES

[1] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. In *Advances in neural information processing systems*, pages 41–48, 2007.

[2] Krishnakumar Balasubramanian, Pinar Donmez, and Guy Lebanon. Unsupervised supervised learning ii: Margin-based classification without labels. *Journal of Machine Learning Research*, 12(Nov):3119–3145, 2011.

[3] John Blitzer, Sham Kakade, and Dean Foster. Domain adaptation with coupled subspaces. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 173–181, 2011.

[4] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[5] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8, 2018.

[6] Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28, 1979.

[7] Pinar Donmez, Guy Lebanon, and Krishnakumar Balasubramanian. Unsupervised supervised learning i: Estimating classification and regression errors without labels. *Journal of Machine Learning Research*, 11(Apr):1323–1351, 2010.

[8] Sujuan Gao. A shared random effect parameter approach for longitudinal dementia data with non-ignorable missing data. *Statistics in Medicine*, 23(2):211–219, 2004.

[9] Pedro J GarcíA-Laencina, José-Luis Sancho-GóMez, and AníBal R Figueiras-Vidal. Classifying patterns with missing values using multi-task learning perceptrons. *Expert Systems with Applications*, 40(4):1333–1341, 2013.

[10] Susan E Hardy, Heather Allore, and Stephanie A Studenski. Missing data: a special challenge in aging research. *Journal of the American Geriatrics Society*, 57(4):722–729, 2009.

[11] José Miguel Hernández-Lobato, Neil Houlsby, and Zoubin Ghahramani. Probabilistic matrix factorization with non-random missing data. In *International Conference on Machine Learning*, pages 1512–1520, 2014.

TABLE VI
THE TOP-10 FEATURES WITH THE HIGHEST VALUE OF WEIGHT IN THE SIX DATASET.

Rank	Philippines			India		
	Petition Field	Feature	Weight Value	Petition Field	Feature	Weight Value
1	petition body	home	0.9897	petition tag	food	2.2064
2	petition description	commission	0.9027	petition tag	environment	1.1792
3	petition title	shut	0.6086	basic properties	calculated goal	1.1250
4	petition title	business	0.5418	petition title	death	0.9560
5	petition body	death	0.5418	petition tag	veterans	0.9538
6	petition description	Party	0.4660	victory description	funding	0.9533
7	petition description	safety	0.4534	petition title	parks	0.9203
8	petition title	tax	0.4443	petition tag	cancer	0.9054
9	petition description	team	0.4397	petition title	dress	0.8802
10	petition body	city	0.3815	petition description	annual	0.8555
Rank	Germany			Australia		
	Petition Field	Feature	Weight Value	Petition Field	Feature	Weight Value
1	petition tag	disability rights	2.3218	petition tag	pedestrian safety	1.8538
2	petition tag	India	2.2784	petition title	forest	1.8490
3	petition tag	criminal justice	2.2518	petition tag	human rights	1.8068
4	petition tag	community development	2.2505	petition title	women	1.6542
5	basic properties	calculated goal	2.0557	petition title	music	1.5052
6	petition title	fire	1.9720	petition tag	wrongful convictions	1.4148
7	petition title	township	1.8220	petition description	ministry	1.3733
8	petition tag	business and brands	1.5872	basic properties	progress	1.3442
9	petition tag	food	1.5858	petition description	time	1.3201
10	petition description	team	1.5570	petition description	section	1.2591
Rank	Canada			United States		
	Petition Field	Feature	Weight Value	Petition Field	Feature	Weight Value
1	petition tag	wrongful convictions	2.1268	victory description	department	3.3964
2	petition tag	india	1.9572	petition tag	k-12	3.0348
3	petition title	centre	1.2078	petition tag	healthcare	2.9356
4	victory description	department	1.1897	petition tag	gay rights	2.7908
5	petition title	traffic	1.1884	petition tag	disability rights	2.7803
6	petition description	federal	1.1563	petition tag	students	2.7134
7	petition title	exam	1.0308	petition tag	clemency	2.6915
8	petition title	island	1.0127	petition tag	education	2.6883
9	petition description	sense	1.0111	victory description	continue	2.6612
10	petition description	license	0.9868	petition tag	cats	2.5314

[12] Bo-Jian Hou, Lijun Zhang, and Zhi-Hua Zhou. Storage fit learning with unlabeled data. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1844–1850, 2017.

[13] Ariel Jaffe, Ethan Fetaya, Boaz Nadler, Tingting Jiang, and Yuval Kluger. Unsupervised ensemble learning with dependent classifiers. In *Artificial Intelligence and Statistics*, pages 351–360, 2016.

[14] Renee Li, Andreas Züfle, Liang Zhao, and Georgios Lamprianidis. Modeling and prediction of people's needs (vision paper). In *Proceedings of the 1st ACM SIGSPATIAL Workshop on Analytics for Local Events and News*, page 3. ACM, 2017.

[15] Yu-Feng Li, Ivor W Tsang, James T Kwok, and Zhi-Hua Zhou. Convex and scalable weakly labeled svms. *The Journal of Machine Learning Research*, 14(1):2151–2188, 2013.

[16] Yu-Feng Li and Zhi-Hua Zhou. Towards making unlabeled data never hurt. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(1):175–188, 2015.

[17] Zachary C. Lipton, David C. Kale, and Randall Wetzel. Modeling missing data in clinical time series with rnns. 2016.

[18] Emmanouil Antonios Platanios, Avrim Blum, and Tom Mitchell. Estimating accuracy from unlabeled data. 2014.

[19] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.

[20] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

[21] Jacob Steinhardt and Percy S Liang. Unsupervised risk estimation using only conditional independence structure. In *Advances in Neural Information Processing Systems*, pages 3657–3665, 2016.

[22] Shuo Xiang, Lei Yuan, Wei Fan, Yalin Wang, Paul M Thompson, and Jieping Ye. Multi-source learning with block-wise missing data for alzheimer's disease prediction. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 185–193. ACM, 2013.

[23] Jianpeng Xu, Pang Ning Tan, Jiayu Zhou, and Lifeng Luo. Online multi-task learning framework for ensemble forecasting. *IEEE Transactions on Knowledge Data Engineering*, PP(99):1–1, 2017.

[24] Lei Yuan, Yalin Wang, Paul M Thompson, Vaibhav A Narayan, and Jieping Ye. Multi-source learning for joint analysis of incomplete multi-modality neuroimaging data. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1149–1157. ACM, 2012.

[25] Jin Wei Zhang, Hui Juan Lu, Wu Tao Chen, and Yi Lu. A comparison study of cost-sensitive learning and sampling methods on imbalanced data sets. *Advanced Materials Research*, 271-273:1291–1296, 2011.

[26] Yu Zhang and Qiang Yang. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017.

[27] Yuchen Zhang, Xi Chen, Denny Zhou, and Michael I Jordan. Spectral methods meet em: A provably optimal algorithm for crowdsourcing. In *Advances in neural information processing systems*, pages 1260–1268, 2014.

[28] Liang Zhao, Qian Sun, Jieping Ye, Feng Chen, Chang-Tien Lu, and Naren Ramakrishnan. Multi-task learning for spatio-temporal event forecasting. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1503–1512. ACM, 2015.

[29] Liang Zhao, Junxiang Wang, Feng Chen, Chang-Tien Lu, and Naren Ramakrishnan. Spatial event forecasting in social media with geographically hierarchical regularization. *Proceedings of the IEEE*, 105(10):1953–1970, 2017.

[30] Liang Zhao, Junxiang Wang, and Xiaojie Guo. Distant-supervision of heterogeneous multitask learning for social event forecasting with multilingual indicators. 2018.

[31] Liang Zhao, Jieping Ye, Feng Chen, Chang-Tien Lu, and Naren Ramakrishnan. Hierarchical incomplete multi-source feature learning for spatiotemporal event forecasting. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2085–2094. ACM, 2016.

[32] Jiayu Zhou, Jianhui Chen, and Jieping Ye. Malsar: Multi-task learning via structural regularization. *Arizona State University*, 2011.

[33] Jiayu Zhou, Jun Liu, Vaibhav A. Narayan, and Jieping Ye. Modeling disease progression via fused sparse group lasso. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1095–1103, 2012.