Online Bayesian Models for Personal Analytics in Social Media

Svitlana Volkova and Benjamin Van Durme
Center for Language and Speech Processing, Johns Hopkins University, Baltimore MD 21218, USA
svitlana@jhu.edu, vandurme@cs.jhu.edu

Abstract

Latent author attribute prediction in social media provides a novel set of conditions for the construction of supervised classification models. With individual authors as training and test instances, their associated content ("features") are made available incrementally over time, as they converse over discussion forums. We propose various approaches to handling this dynamic data, from traditional batch training and testing, to incremental bootstrapping, and then active learning via crowdsourcing. Our underlying model relies on an intuitive application of Bayes rule, which should be easy to adopt by the community, thus allowing for a general shift towards online modeling for social media.

Introduction

The recent explosion of social media has led to an interest in predicting hidden information from the large amounts of freely available content. As compared to the earlier explosion of documents arising from the web, social media content is significantly more *personalized*, i.e., written in the first person, informal, and often revealing of latent properties of the author. This has become known alternatively as constructing: user demographic models, personal analytics or customer profiling services. Researchers have explored the prediction of latent attributes including gender (Rao et al. 2010; Filippova 2012; Bergsma et al. 2013), age (Nguyen et al. 2013), political preferences (Conover et al. 2011b; Cohen and Ruths 2013; Volkova, Coppersmith, and Van Dume 2014), personality traits (Bachrach et al. 2012), and so on.

However, the majority of this work treats the modeling task much as prior work on non-social media: construct a corpus of labeled materials, and perform supervised classification in a batch setting. This ignores one of the primary distinguishing characteristics of social media content: it is dynamically generated over time, and usually centered within the context of a social network (i.e., friends or other types of associates of the author). Further, different users of the medium contribute to greater or lesser extent: a given user may send one tweet a week, or one tweet an hour, etc. Prior work tends to gloss over this fact by building controlled collections with a large, fixed amount of content assumed per user e.g., 1K tweets. (Zamal, Liu, and Ruths 2012) or even 5K tweets (Rao et al. 2010).

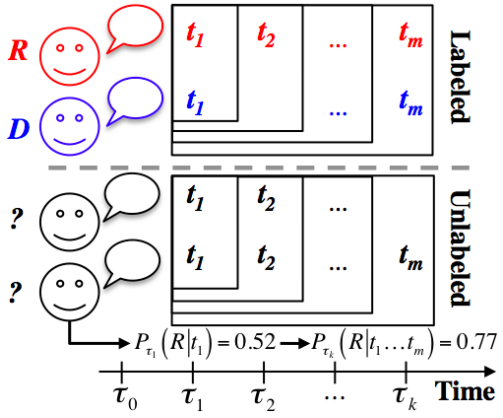


Figure 1: An example of political preference prediction over a dynamic stream of communications. R stands for Republican and D for Democratic users. As time τ goes by, both labeled and unlabeled users generate tweets $t_1 \ldots t_m$. Boxes outline the amount of train and test data available at each τ_k.

In contrast, Burger et al. and Volkova, Coppersmith, and Van Dume showed the intuitive importance of the amount of content available per user at test time: the more content you have, the better your predictions. This was followed by Van Durme who proposed a model that allowed for incremental updating of classifier predictions over time, as users continued to author new content. This model treated each user as a sort of dynamic feature vector that evolved over time, and assumed access to a pre-trained classification model based on labeled data available *a priori*, akin to earlier work in the purely batch setting.

Here we go beyond the existing work and propose two novel contributions in mining streaming social media: (1) contrasting Van Durme, we treat each new message as independent evidence which is combined into an incremental user-prediction model as a straightforward application of Bayes Rule; (2) we explore model training in parallel to its application, rather than assuming a previously existing labeled dataset. Also, distinct from Van Durme, but previously explored in the batch-setting by (Zamal, Liu, and Ruths 2012) and (Volkova, Coppersmith, and Van Dume 2014) we make use of the local user neighborhood in our dynamic model.

Our approach captures the same incremental intuitions as the work by Van Durme, but we situate it within the well understood framework of Bayesian inference: we hope this will encourage others to build upon this effort in constructing more complicated models. Further, by recognizing that both *training* as well as *testing* materials are dynamically generated in social media, then possibly coupled to dynamic model feedback via crowdsourcing, this suggests latent author attribute prediction as a rich source for methodological challenges in online and active learning. This work means to give perspective on the various ways this dynamism may be incorporated into an experimental framework. Future work may then choose a particular path and focus on models with further complexity and larger datasets.

Approach

Data

Our approach is relevant generally to multi-class prediction problems in social media. Here we focus on a binary prediction task, specifically the prediction of political preference as captured by the dominant two American political parties: Democratic and Republican. We rely on a dataset previously used for political affiliation classification by (Pennacchiotti and Popescu 2011), then (Zamal, Liu, and Ruths 2012) and (Volkova, Coppersmith, and Van Dume 2014).[1] The original data consists of 200 Republican and 200 Democratic users associated with 925 tweets on average per user. Each user has on average 6155 friends with 642 tweets per friend. Sharing restrictions[2] and rate limits on Twitter data collection only allowed us to recreate a subset of that collection. Based on the subset we were able to obtain we formed a balanced collection of 150 Democratic and 150 Republican users. For each user, we randomly sampled 20 friends with 200 tweets per friend.

Models

We assume a set of independent users $U = \{u_i\}$, and neighbors $N = \{n_j\}$, with $N^{(u)}$ the neighbors of u.[3] We are concerned with models over data that changes over time: let τ be an index over discrete time-steps, where at each time-step τ_k we observe zero or more tweets from each user, and each user-neighbor, on which we base our predictions. A user is *labeled* at time τ if we know the value of the attribute function $\mathrm{A}(u) \in \{a_l\}$.

[1]The original Twitter users with their political labels extracted from http://www.wefollow.com as described by (Pennacchiotti and Popescu 2011). The user-friend data was collected by (Zamal, Liu, and Ruths 2012) and expanded with other neighbors by (Volkova, Coppersmith, and Van Dume 2014). User/tweet IDs and user-friend relations can be found at http://www.cs.jhu.edu/∼svitlana/.

[2]Twitter only allows to share user and tweet IDs. The actual content e.g., tweets or user meta data can be download by querying Twitter API. However, as of Aug. 2013, a certain portion of user profiles were deleted or became private: this is a standard issue in reproducing prior results on Twitter and is not specific to this work. Moreover, note that Twitter API restricts queries to 720 per hour.

[3]In our experiments those neighbors will be the *friends* of a user on Twitter.

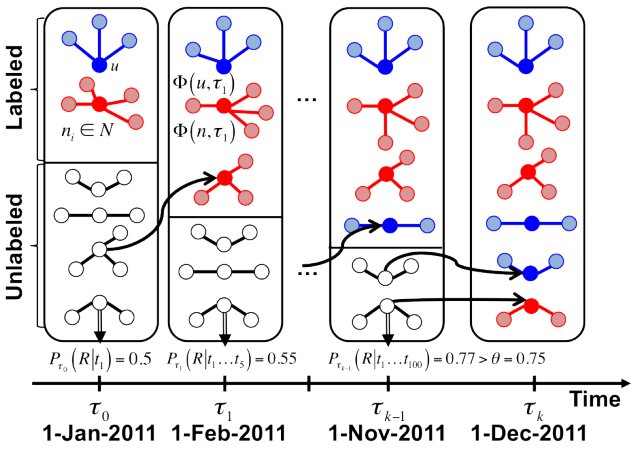


Figure 2: Active learning classification setup: nodes represent Twitter users, edges stand for friend relationships between the users; dark red and blue nodes represent labeled R and D users; light red and blue nodes represent friends of R and D users; $\Phi_U(\tau_1)$ and $\Phi_N(\tau_1)$ are the models trained exclusively on user or neighbor (friend) content.

For example, in our experiments we will model the (American) political preference attribute, defined as: $\mathrm{A}(u) \in \{\mathtt{R}, \mathtt{D}\}$, with R standing for Republican and D for Democratic. Let $L_\tau \subseteq U$ be the labeled users at time τ, and $\overline{L_\tau} = U \backslash L_\tau$ the unlabeled users. Our goal is to predict the attribute value for each user in $\overline{L_\tau}$ at every τ given the evidence available up to τ.

Unlike previous models for latent user attribute classification, we: (1) consider updating the initial model learned at τ_0 as new evidence becomes available at τ_k, and (2) reestimate decision probabilities for the unlabeled users given the updated model and new content generated by these users and their neighbors by τ_k.

We define two models $\Phi(u,\tau)$ and $\Phi(n,\tau)$ learned from dynamically growing streams of tweets $T^{(U)}$ and $T^{(N)}$. The user model $\Phi(u,\tau)$ is learned exclusively from user communications to be applied to user tweets $t_1^{(u)}, t_2^{(u)}, \ldots, t_m^{(u)} \in T_\tau^{(u)}$. $\Phi(u,\tau)$ is then a function mapping a user to the most likely attribute value assignment at τ:

$$\Phi(u,\tau) = \mathrm{argmax}_a \mathrm{P}(\mathrm{A}(u) = a | T_\tau^{(u)}). \quad (1)$$

Neighbor model $\Phi(n,\tau)$ is learned from neighbor communications of Democratic and Republican users. It is defined similarly to Eq. 1 and is applied to classify friend tweets within friend or joint user-friend stream $t_1^{(n)}, t_2^{(n)}, \ldots, t_m^{(n)} \in T_\tau^{(n)}$.

A user is labeled at time τ if we predict the value of the attribute function $\mathrm{A}(u)$. We apply Bayesian rule updates to dynamically revise posterior probability estimates of the attribute value $\mathrm{P}(\mathrm{A}(u) = \mathtt{R} \mid T_\tau)$ given a prior e.g., in our case we start with a balanced prior $\mathrm{P}(\mathtt{R}) = \mathrm{P}(\mathtt{D}) = 0.5$.

$$\mathrm{P}(\mathrm{A}(u) = \mathtt{R} \mid T_\tau) = \frac{\mathrm{P}(\mathrm{A}(u) = \mathtt{R}) \cdot \mathrm{P}(T_\tau | \mathrm{A}(u) = \mathtt{R})}{\sum_{a \in \mathrm{A}} \mathrm{P}(\mathrm{A}(u) = a) \cdot \mathrm{P}(T_\tau | \mathrm{A}(u) = a)}. \quad (2)$$

We will assume tweets to be independent conditioned on attribute, which means our model factors across individual messages $T_\tau = (t_1, \ldots, t_m)$, allowing for simple posterior updates on a tweet by tweet basis:

$$\mathrm{P}(\mathrm{A}(u) = \mathtt{R} \mid t_1 \ldots t_m) = \frac{\mathrm{P}(\mathrm{A}(u) = \mathtt{R}) \cdot \prod_m \mathrm{P}(t_m|\mathrm{A}(u) = \mathtt{R})}{\sum_{a \in \mathrm{A}} \mathrm{P}(\mathrm{A}(u) = a) \prod_m \mathrm{P}(t_m|\mathrm{A}(u) = a).} \quad (3)$$

The conditional probability of a given tweet is determined by a log-linear model trained on observations from L_τ. We show the example updated posterior probabilities for political preference prediction $\mathrm{P}(\mathtt{R} \mid t_1 \ldots t_m)$ in Figure 2.

The final decisions about label assignments can be made at any time τ_k e.g., if $\mathrm{P}(\mathtt{R} \mid t_1 \ldots t_m) = 0.9$ one can label the user as R with an associated 90% model confidence given the evidence available by τ_k. We analyze the difference in precision and recall by making decisions based on high or low probability assignments using different thresholds θ: 0.55 and 0.95. When $\mathrm{P}(\mathrm{A}(u) = a \mid t_1 \ldots t_m)$ exceeds θ we make a decision about the label for a user at τ_k.

Experimental Setup

We design a set of classification experiments from three types of data streams including user (U), neighbor (N) and user-neighbor (UN). We aim to explore the following prediction settings: Iterative Batch (IB), Iterative Batch with Rationale Filter (IBR), Active without Oracle (AWOO), Active with Oracle (AWO), Active with Rationale Filter (AWR).

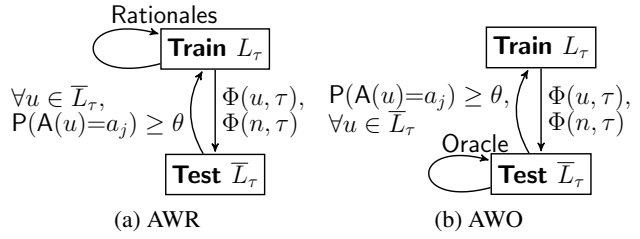
For all settings we perform 6-fold cross validation and use a balanced prior:[4] 50 users in the train split and 250 users in the test. For all experiments we use the LIBLINEAR package integrated in the JERBOA toolkit (Van Durme 2012a). The log-linear models with dynamic Bayesian updates defined in Eq.1 and Eq.3 are learned using binary word unigram features extracted from user or neighbor content.

Iterative Batch We learn tweet-based models at each time stamp τ from the set of labeled users L_τ and their neighbors e.g., friends. We apply these models using Eq.3 to U, N and UN streams to label all unlabeled users $\overline{L}_\tau$ over time. The set of labeled users is constant across all values of τ: we have labels on some users before hand, and no new labels are gathered; only the amount of content available for the users and their neighbors is increasing over time.

Iterative Batch With Rationale Filtering Prior work by (Zaidan and Eisner 2008) and (Yessenalina, Choi, and Cardie 2010) explored the utility of asking annotators to choose *rationales*, explicitly highlighted words or phrases in provided content, that best justified why the annotator made their labeling decision. Our batch setup with rationale filtering is equivalent to the iterative batch setup, except at every time stamp τ we modify our training data to include tweets with the rationales exclusively. For that, at every τ we estimate predictive unigrams – potential rationale words $w \in V$ for Democratic and Republican users in L_τ:

$$V^{a \in \mathrm{A}(u)} = \{w \mid \mathrm{P}(w \mid \mathrm{A}(u) = a) \geq 0.55\} \quad (4)$$

[4]Our framework generalizes to non-balanced class priors for train and test, but does assume that the prior is known *a priori*; estimating class priors in social media is an element of future work.



(a) AWR (b) AWO

Figure 3: Active setting with: (a) rationales and (b) oracle. Active setting without oracle (AWOO) is similar to (a) AWR except the rationale filtering step is omitted.

The conditional probabilities of each word $\mathrm{P}(w|\mathrm{A}(u) = \mathtt{D})$ and $\mathrm{P}(w|\mathrm{A}(u) = \mathtt{R})$ are calculated as the empirical estimates over tweets, where w was constrained to have a minimum count of three. We then ask annotators on Mechanical Turk to select rationales from the strongest ranked candidates for D and R by showing them potential rationales and a subset of tweets up to τ. We ask three redundant annotations for each unigram w and take the majority vote to determine if the unigram truly reveals political preferences. For example, the Democratic rationales with $\mathrm{P}(w \mid \mathrm{A}(u) = a) > 0.9$ and 100% annotator agreement include: *immigrants, students, unemployment* and Republican: *#teaparty, dental, obamacare, arts*.[5]

Active Without Oracle Unlike our batch setup applied iteratively over a stream of tweets, we propose to update the $\Phi(u,\tau)$ and $\Phi(n,\tau)$ models by moving users from the test set labeled at τ_k to the training set at τ_{k+1} as shown in Figure 2. The final decisions about class labels for the unlabeled users are made based on posterior probability estimates $\mathrm{P}(\mathrm{A}(u) \mid T_\tau)$ to exceed the threshold θ. Similarly to the batch setting we experiment with two values of θ and three data streams: U, N and UN. This bootstrapping approach we refer to as active without oracle (AWOO).

Active With Oracle Alternatively, the final label assignments can be judged by an oracle e.g., annotators on Mechanical Turk. For example, we might show m tweets produced by the user by time τ to one or more annotators. And only if one or more independent annotator judgments agree with $\Phi(u,\tau)$, then we assign a corresponding label to this user at τ_k, and move this user to the training set at τ_{k+1}. Here, since we know the labels we simulate turker judgments (so the oracle is 100% correct). Thus, this setup measures the upper bound for classification. But in the future, we would like to engage real turkers to make class label judgments in the loop. We refer to this setup as active with oracle (AWO) and show it in Figure 3b.

Active With Rationale Filtering The rationale filtering step used for IBR setup is also applied to AWOO setup at every τ as shown in Figure 3a. The difference between batch and active models with rationale filtering is that the potential rationales are estimated on a different set of training data using Eq. 4. In the active case tweets from previously unlabeled users that exceed θ at τ_k are added to the tweets of labeled users at τ_{k+1}.

[5]Complete rationale lists for political preference as well as other attributes e.g., gender and age can be downloaded from http://www.cs.jhu.edu/∼svitlana/rationales.html

Evaluation

We are concerned with *accuracy* when operating at different confidence thresholds. Let $\text{Acc}_{\tau,\theta}$ be the accuracy at τ, when considering just users for which the posterior probability exceeds θ. At a given value of τ and θ, let:

$$\text{Acc}_{\tau,\theta} = \frac{TR + TD}{R + D}, \quad (5)$$

where TR = true Republicans, TD = true Democrats, and R, D are the number of users labeled as Republicans or Democrats, respectively.[6] We abbreviate this as: $\text{Acc}_{\tau,\theta} = C_{\tau,\theta}/A_{\tau,\theta}$, with $C_{\tau,\theta}$ being the number of *correctly* classified users, and $A_{\tau,\theta}$ being the number of users *above* a given threshold θ. We also estimate $Q_{\tau,\theta}$ which is the total number of active users who tweeted at least once by τ (note that $C_{\tau,\theta} \leq A_{\tau,\theta} \leq Q_{\tau,\theta}$). The performance metric $\text{Acc}_{\tau,\theta}$ defined in Eq. 5 can be effectively used for targeted online advertising where one would like to send the advertisements as early as possible to only active users at time τ for whom labels are assigned with a reasonable confidence θ.

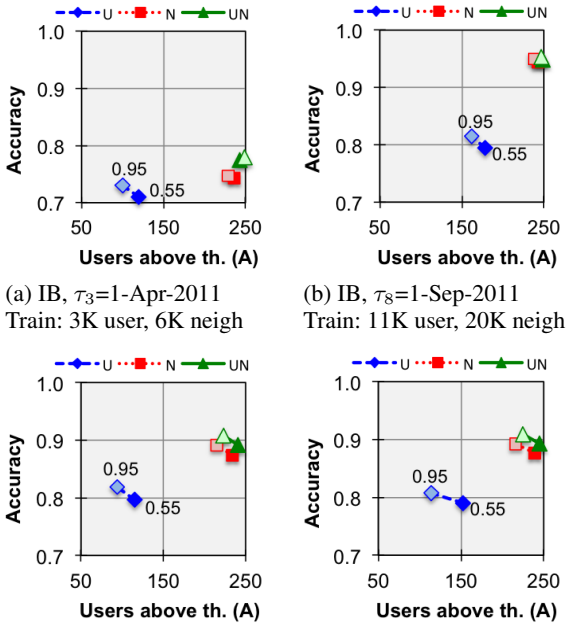
Results

We first confirm that our incrementally batch-trained approach performs as would be expected. In Figure 4 (a - b), consider model U (based only on user tweets): the difference between decision thresholds 0.95 and 0.55 shows a classic precision versus recall tradeoff; at 0.95 less users are classified (x-axis) but at higher precision (y-axis), as compared to 0.55 which instead has higher recall. This pattern repeats for all models U, N and UN, trained and tested with less data (a: Jan - Apr) as well as more data (b: Jan - Sep). With more data (b), performance improves for all scenarios. U is outperformed by N and UN: having access to the content of neighbors improves performance considerably in all cases (affirming the conclusions of (Zamal, Liu, and Ruths 2012) and (Volkova, Coppersmith, and Van Dume 2014)).

Next we contrast those results to AWOO: not only do we retrain the model each month as in batch, but now we bootstrap by taking our most confident (0.95 or 0.55) predictions for users and add them into our labeled set as if their labels were known. We found that our AWOO model yields higher performance than IB model in early months (up to 1-Jul-2011), and insignificantly lower results after that. It happens because in the active setting the model accumulates noisy predictions for some users over time. In contrast, the AWO model does not have this issue and yields consistently better results over time as we show latter. In Figure 5 we present more detailed classification results for batch and active setting for two thresholds 0.55 and 0.95. These results allow us to analyze (a) the threshold and (b) data stream type influence on classification performance as shown below.

Analyzing Threshold Influence

The results in Figures 4 and 5 demonstrate that for higher θ, when the models are more constrained and, therefore, more confident about their predictions, less users A are above the threshold θ. Consequently, the number of correctly classified users C is lower for 0.95 compared to 0.55. Therefore, one has to make a decision about θ taking into account this precision-recall tradeoff: models with higher θ are more precise but yield lower recall vs. models with lower θ are less precise but yield higher recall over time τ.



(a) IB, τ_3=1-Apr-2011 Train: 3K user, 6K neigh

(b) IB, τ_8=1-Sep-2011 Train: 11K user, 20K neigh

(c) AWOO, τ_3=1-Apr-2011 Train: 18K user, 34K neigh

(d) AWOO, τ_8=1-Sep-2011 Tr: 61K user, 125K neigh

Figure 4: The comparison of batch (IB) vs. active (AWOO) setting using U, N and UN data streams and different confidence thresholds: $\theta = 0.55$ (bold) and $\theta = 0.95$ (light) markers.

Moreover, for our active setting threshold θ has another important objective – to control the amount and quality of the data labeled at τ_k and used to update the model at τ_{k+1}. The results in Figure 5 show that the active models outperform the iterative batch models in terms of recall in early months. This results are very important for targeted advertising scenario when more ads need be sent to more users as early as possible.

Studying Data Stream Type Influence

We observe that in all settings when the probability estimates are updated from N, UN streams compared to U stream the # of correctly classified users $C_{\tau,\theta}$ at each τ is significantly higher. The reason for UN, N streams yielding better results is that more tweets associated with the user e.g., friend tweets that carry a substantial signal for prediction become available.[7] However, relative gains over time for N and UN are lower compared to U stream. It is because "less difficult to classify" users are easily classified using UN (N) streams earlier at τ_k and only "more difficult to classify" users are left to be classified later at τ_{k+1}.

[6] This generalizes standard language of (True) Positive and (True) Negative to allow for non-binary scenarios, such as if adding Libertarian (L), Green Party (G), etc., to the attribute set: Acc$=(TR + TD + TG + TL)/(R + D + G + L)$.

[7] Many authors don't tweet often e.g., 85.3% of all Twitter users post less than one update per day. Thus, less tweets are generated by random users by time τ compared to the number of tweets generated by a set of their friends.

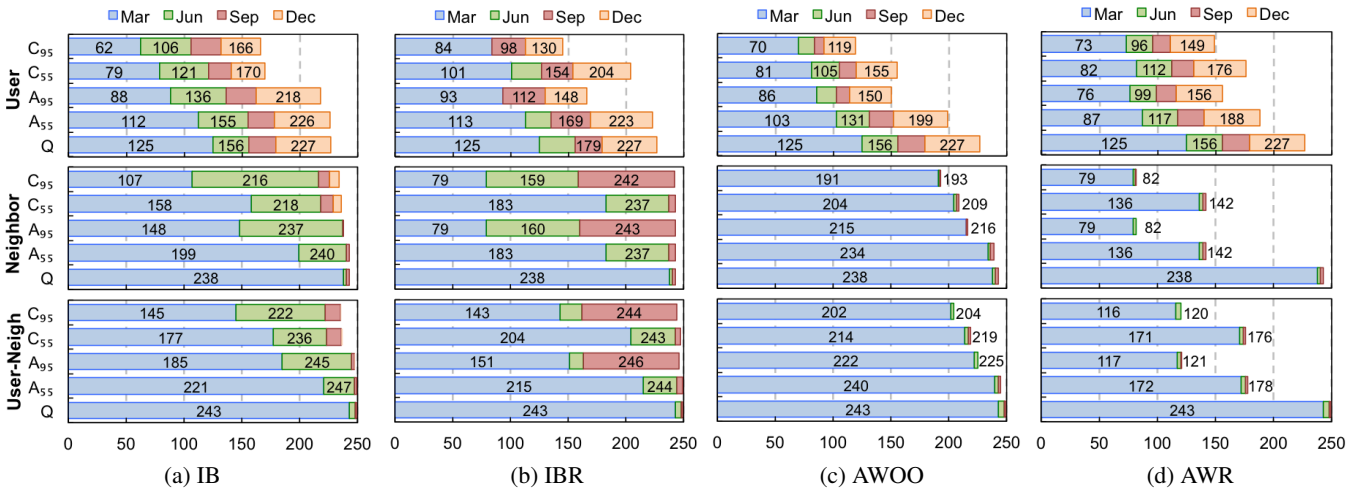


Figure 5: Comparing iterative batch IB, IBR vs. active models AWOO, AWR with and without rationale filtering for political preference prediction with two thresholds $\theta = 0.55$ and $\theta = 0.95$ applied to user, neighbor and user-neighbor communication streams. Starting on 1-Jan-2011, at each time stamp τ e.g., τ_2=1-Mar-2011, ..., τ_{11}=1-Dec-2011 we measure $C_{\tau,\theta}$ = the # of correctly classified users, $A_{\tau,\theta}$ = the # of users above the threshold θ, Q_τ = the # of users who tweeted at least once by time τ.

Utilizing Oracle Annotations

In Figure 6 we demonstrate the upper bound for political preference classification performance with $\theta = 0.95$ using our active with oracle (AWO) experimental setup. Similar to other experiments, we report classification performance $\text{Acc}_{\tau,\theta}$ at every τ with the number of user and neighbor tweets available for training when predictions are made over U and N data streams. We find that $\text{Acc}_{\tau,\theta}$ is monotonically increasing over time and is significantly higher then for IB and AWOO settings. To give a cost estimate of requesting iterative oracle annotations, we outline the number of requests to the oracle aggregated over time in Figure 6 (top).

Active learning with iterative oracle annotations demonstrate the highest performance compared to all other classification settings. For instance, 226 out of 250 users (90%) are correctly classified by June using N stream and 230 (92%) using UN stream using AWO setup compared to 191 (76%) and 203 (81%) users using AWOO setup. Similarly, 112 (45%) users are correctly classified by June using U stream using AWO model compared to 80 (32%) using AWOO.

Applying Rationale Filtering

Here we analyze the impact of rationale filtering on prediction performance in batch: IB vs. IBR and active: AWOO vs. AWR settings over time. In Figure 5 we report results for models with and without rationale filtering. As before, we present the results for two thresholds 0.55 and 0.95 and three data streams: U, N and UN. For IBR and AWR models with rationale filtering we observe similar precision-recall trends to IB and AWOO models shown in Figure 4.

During rationale filtering we select training examples with highly predictive norms (a.k.a. rationales) at every τ. This filtering step reduces the number of training examples L, vocabulary size V and feature space for both user $\Phi(u, \tau)$ and neighbor $\Phi(n, \tau)$ models over time as shown in Tables 1 and 2. We observe that the size of the training data L is reduced at least in half at every time stamp. Therefore, we consider

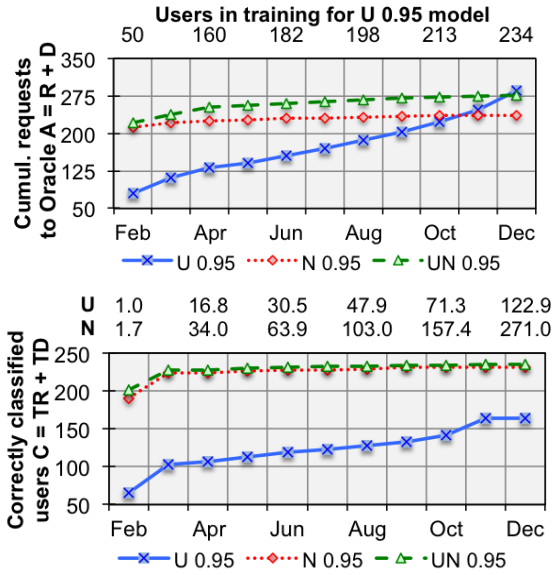


Figure 6: Active with oracle (AWO) classification results using U, N and UN streams and $\theta = 0.95$ (**U** and **N** stand for thousands of tweets used to train user and neighbor models).

rationale filtering as a dimensionality reduction step for our batch and active models with incremental Bayesian updates.

Nevertheless, the size of the training data is significantly lower at every τ the quality of batch and active models trained with filtered data is better for IBR vs. IB and AWR vs. AWOO. In other words, selecting tweets with highly predictive feature norms for training leads to consistent performance improvements over time. We show the empirical results for the relative percentage gain $\Delta\text{Acc}, \%$ for batch and active models with vs. without rationale filtering in Tables 1 and 2, respectively. Models with rationale filtering yield higher precision but lower recall compared to the models without rationale filtering when the predictions are made using N or UN stream. Except we observe higher precision but comparable or higher recall when U stream is used.

		Mar	Jun	Sep	Dec
U	$\Delta\text{Acc}_{0.55}$	+21.7	+20.9	+14.2	+21.6
	$\Delta\text{Acc}_{0.95}$	+27.7	+18.5	+7.8	+15.1
	ΔL_θ	1.1	2.4	4.2	9.0
	ΔV_θ	0.8	1.3	1.9	3.0
N	$\Delta\text{Acc}_{0.55}$	+5.6	+10.6	+5.8	+3.0
	$\Delta\text{Acc}_{0.95}$	+13.2	+11.7	+5.6	+3.0
	ΔL_θ	2.5	6.9	12.5	24.9
	ΔV_θ	1.7	3.4	4.8	7.0
UN	$\Delta\text{Acc}_{0.55}$	+19.2	+9.9	+4.4	+10.3
	$\Delta\text{Acc}_{0.95}$	+21.5	+9.1	+4.0	+10.8

Table 1: The difference between IBR and IB settings.

		Mar	Jun	Sep	Dec
U	$\Delta\text{Acc}_{0.55}$	+20.6	+19.7	+18.5	+20.1
	$\Delta\text{Acc}_{0.95}$	+19.0	+18.7	+18.0	+20.6
	$\Delta L_{0.95}$	3.5	11.9	16.3	32.4
	$\Delta V_{0.95}$	2.5	3.8	6.3	11.2
N	$\Delta\text{Acc}_{0.55}$	+14.1	+13.9	+13.7	+13.7
	$\Delta\text{Acc}_{0.95}$	+11.7	+11.7	+11.9	+11.9
	$\Delta L_{0.95}$	7.9	19.1	36.5	81.4
	$\Delta V_{0.95}$	2.7	4.6	7.3	13.0
UN	$\Delta\text{Acc}_{0.55}$	+11.5	+10.8	+10.7	+10.7
	$\Delta\text{Acc}_{0.95}$	+9.5	+9.4	+9.4	+9.4

Table 2: AWR vs. AWOO settings: ΔAcc_θ represents relative percentage gain between AWR and AWOO, ΔL_θ is the difference in the # of tweets available for training, ΔV_θ is the difference in feature space (vocabulary) size at τ_k.

To summarize, rationale filtering significantly improves classification accuracy and can be effectively used for attribute prediction that require high precision e.g., product likes or personal interests. For batch setting, IBR setup yields much better results than IB setup as high as $\text{Acc}_{\text{Mar},0.95} = 27.7\%$. For active setting, AWR setup yields as high as $\text{Acc}_{\text{Mar},0.55} = 20.6\%$ gain over AWOO using U stream. Moreover, for both batch and active setting: the higher $\Delta\text{Acc}_{\tau,\theta}$ reported when predictions are made from U compared to N or UN streams; the incremental relative gains for $\text{Acc}_{\tau,\theta}$ are higher for 0.55 compared to 0.95 models.

Applications

Our approaches for making predictions over dynamically evolving social media streams based on incremental Bayesian online updates can be effectively used in: (1) real-time streaming scenarios for dynamically growing social networks; (2) limited resource training conditions e.g., iterative retraining and active learning (bootstrapping) will allow exploring new understudied attributes e.g., life satisfaction, relationship status for which no or limited labeled data exists; (3) low-resource prediction settings e.g., when no or limited user data is available at any given time, neighbor streams can be used to make predictions about the user; (4) low-cost annotation models that rely on iterative *instance* (assigning class labels to users) or *feature* annotations (highlighting predictive words in tweets) via crowdsourcing. Moreover, our batch and active models with iterative rationale filtering help to reduce storage and memory requirements when processing large feature vectors and iterative re-training models for real-world prediction in social media.

Related Work

Batch Models for Personal Analytics The vast majority of works on predicting latent user attributes in social media apply supervised batch models trained on large amounts of user generated content except (Burger et al. 2011; Volkova, Coppersmith, and Van Dume 2014). These models are learned using lexical bag-of-word features for classifying user: gender (Rao et al. 2010; Filippova 2012; Ciot, Sonderegger, and Ruths 2013; Bergsma and Van Durme 2013); age (Goswami and Shishodia 2012; Nguyen et al. 2013; Schwartz et al. 2013a); political orientation (Conover et al. 2011a; Pennacchiotti and Popescu 2011; Zamal, Liu, and Ruths 2012; Cohen and Ruths 2013); personality (Golbeck et al. 2011; Bachrach et al. 2012; Schwartz et al. 2013b). Some works use unsupervised approaches to learn user demographics in social media including large-scale clustering (Bergsma et al. 2013) and probabilistic graphical models (Eisenstein et al. 2010).

Streaming Models for Personal Analytics Van Durme proposed a streaming model with incremental updates for iteratively predicting user attributes over a stream of communications (Van Durme 2012b). Unlike Van Durme's approach, our model suggests more straightforward application of incremental tweet-level online Bayesian updates. In addition, we explore batch vs. online retraining for incremental updates of our models. Finally, we take advantage of Twitter network structure and experiment with friend streams in addition to the user stream of communications. Moreover, our active models with iterative oracle and rationale annotations are similar to active learning techniques where the learner is in control of the data used for learning as described in details here (Olsson 2009; Settles 2010; Laws 2012; Settles 2012).

Summary

We proposed novel approaches for making predictions over dynamically evolving social media streams based on incremental Bayesian online updates. We studied an iterative incremental retraining in batch and active settings with and without iterative oracle annotations. Moreover, we applied interactive feature annotation (rationale) technique as a filter for iterative retraining of the proposed models. Finally, we took advantage of a network structure by making predictions from neighbor and joint user-neighbor streams.

Our key findings include: I. Active retraining with correctly classified users from test data added to the training data at every time stamp significantly outperforms iterative batch retraining setup. II. Making predictions using a joint user-neighbor or neighbor stream is more effective than using only user stream. III. Models with higher confidence yield higher precision and models with lower confidence yield higher recall for both batch and active setting. IV. Rationale annotation and filtering during iterative retraining leads up to 27.7% relative improvement in iterative batch and 20.6% in active setting. V. Active retraining with oracle annotations yields the highest recall: 85% of test users are correctly classified after the second iteration using a joint user-neighbor stream.

References

Bachrach, Y.; Kosinski, M.; Graepel, T.; Kohli, P.; and Stillwell, D. 2012. Personality and patterns of facebook usage. In *Proceedings of the 3rd Annual ACM Web Science Conference*, WebSci '12, 24–32.

Bergsma, S., and Van Durme, B. 2013. Using conceptual class attributes to characterize social media users. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, 710–720.

Bergsma, S.; Dredze, M.; Van Durme, B.; Wilson, T.; and Yarowsky, D. 2013. Broadly improving user classification via communication-based name and location clustering on Twitter. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 1010–1019.

Burger, J. D.; Henderson, J.; Kim, G.; and Zarrella, G. 2011. Discriminating gender on Twitter. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1301–1309.

Ciot, M.; Sonderegger, M.; and Ruths, D. 2013. Gender inference of twitter users in non-english contexts. In *EMNLP*, 1136–1145.

Cohen, R., and Ruths, D. 2013. Classifying Political Orientation on Twitter: It's Not Easy! In *Proceedings of the International AAAI Conference on Weblogs and Social Media (ICWSM)*, 91–99.

Conover, M. D.; Ratkiewicz, J.; Francisco, M.; Gonc, B.; Flammini, A.; and Menczer, F. 2011a. Political polarization on Twitter. In *Proceedings of Fifth International AAAI Conference on Weblogs and Social Media (ICWSM)*, 89–96.

Conover, M. D.; Gonçalves, B.; Ratkiewicz, J.; Flammini, A.; and Menczer, F. 2011b. Predicting the political alignment of Twitter users. In *Proceedings of Social Computing*, 192–199.

Eisenstein, J.; O'Connor, B.; Smith, N. A.; and Xing, E. P. 2010. A latent variable model for geographic lexical variation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1277–1287.

Filippova, K. 2012. User demographics and language in an implicit social network. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 1478–1488.

Golbeck, J.; Robles, C.; Edmondson, M.; and Turner, K. 2011. Predicting personality from twitter. In *Proceedings of SocialCom/PASSAT*.

Goswami, S., and Shishodia, M. S. 2012. A fuzzy based approach to stylometric analysis of blogger's age and gender. In *HIS*, 47–51.

Laws, F. 2012. *Effective Active Learning for Complex Natural Language Processing Tasks*. Ph.D. Dissertation, University of Stuttgart.

Nguyen, D.; Gravel, R.; Trieschnigg, D.; and Meder, T. 2013. "How old do you think I am?" A study of language and age in Twitter. In *Proceedings of the AAAI Conference on Weblogs and Social Media (ICWSM)*, 439–448.

Olsson, F. 2009. A literature survey of active machine learning in the context of natural language processing. Technical Report 06.

Pennacchiotti, M., and Popescu, A. M. 2011. A machine learning approach to Twitter user classification. In *Proceedings of the International AAAI Conference on Weblogs and Social Media (ICWSM)*, 281–288.

Rao, D.; Yarowsky, D.; Shreevats, A.; and Gupta, M. 2010. Classifying latent user attributes in Twitter. In *Proceedings of the 2nd International Workshop on Search and Mining User-generated Contents (SMUC)*, 37–44.

Schwartz, H. A.; Eichstaedt, J. C.; Kern, M. L.; Dziurzynski, L.; Ramones, S. M.; Agrawal, M.; Shah, A.; Kosinski, M.; Stillwell, D.; Seligman, M. E.; et al. 2013a. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PloS one* 8(9):e73791.

Schwartz, H. A.; Eichstaedt, J. C.; Dziurzynski, L.; Kern, M. L.; Blanco, E.; Kosinski, M.; Stillwell, D.; Seligman, M. E.; and Ungar, L. H. 2013b. Toward personality insights from language exploration in social media. In *AAAI Spring Symposium: Analyzing Microtext*.

Settles, B. 2010. Active learning literature survey. Technical report, University of Wisconsin–Madison.

Settles, B. 2012. *Active Learning*. Morgan and Claypool.

Van Durme, B. 2012a. Jerboa: A toolkit for randomized and streaming algorithms. Technical report, Human Language Technology Center of Excellence.

Van Durme, B. 2012b. Streaming analysis of discourse participants. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 48–58.

Volkova, S.; Coppersmith, G.; and Van Dume, B. 2014. Inferring user political preferences from streaming communications. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Yessenalina, A.; Choi, Y.; and Cardie, C. 2010. Automatically generating annotator rationales to improve sentiment classification. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, 336–341.

Zaidan, O. F., and Eisner, J. 2008. Modeling annotators: A generative approach to learning from annotator rationales. In *Proceedings of EMNLP 2008*, 31–40.

Zamal, F. A.; Liu, W.; and Ruths, D. 2012. Homophily and latent attribute inference: Inferring latent attributes of Twitter users from neighbors. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 387–390.