Vuvuzelas & Active Learning for Online Classification

Ulrich Paquet
ulripa@microsoft.com

Jurgen Van Gael
jvangael@microsoft.com

David Stern
dstern@microsoft.com

Gjergji Kasneci
gjergjik@microsoft.com

Ralf Herbrich
rherb@microsoft.com

Thore Graepel
thoreg@microsoft.com

Abstract

Many online service systems leverage user-generated content from Web 2.0 style platforms such as Wikipedia, Twitter, Facebook, and many more. Often, the value lies in the freshness of this information (e.g. tweets, event-based articles, blog posts, etc.). This freshness poses a challenge for supervised learning models as they frequently have to deal with previously unseen features.

In this paper we address the problem of online classification for tweets, namely, how can a classifier be updated in an online manner, so that it can correctly classify the latest "hype" on Twitter? We propose a two-step strategy to solve this problem. The first step follows an active learning strategy that enables the selection of tweets for which a label would be most useful; the selected tweet is then forwarded to Amazon Mechanical Turk where it is labeled by multiple users. The second step builds on a Bayesian corroboration model that aggregates the noisy labels provided by the users by taking their reliabilities into account.

1 Introduction

In recent years, there has been an explosion of content creation and information sharing platforms such as Wikipedia, Twitter and Facebook. These platforms allow a large number of experts to share their knowledge and opinions in real time on the latest news, events, and hypes. Many online services aim to make sense of this kind of content by automatically annotating, classifying, or personalizing the information in it. At FUSE Labs [1] we have developed a personalized news recommendation system, called Project Emporia [2], which aggregates content that is published through Twitter and public RSS feeds to build a personalized newspaper for our users. A key component in Project Emporia's pipeline is a classification system, which automatically categorizes tweets with respect to their topics.

A pressing challenge for our classification system is that it must adapt over time to take advantage of new (previously unseen) features. For example, the word "vuvuzela" was virtually unknown on Twitter before the 2010 World Cup. During the World Cup, the word "vuvuzela" became a trending topic and a very indicative feature for the category "sport". Since tweets are limited to 140 characters, a single word can often be a crucial feature for correct classification. Because Project Emporia runs continuously and aims at showing the most relevant news in the last 24 hours, there is a need for fast updates of the classification system, so that users can be provided with the desired experience.

In this paper, we propose a two-step strategy for fast online updates to our classifier. The first step is an online active learning strategy for selecting the next tweet to be labeled. More specifically, as a tweet comes down the Twitter firehose (the stream of *all* public status updates from Twitter), the system decides in real time whether the tweet should be labeled through the classifier or forwarded to Amazon Mechanical Turk (AMT), where AMT workers can label it. In the latter case, the tweet is batched up with other tweets (which the system decided should be labeled) and sent to AMT. Section 3 describes the mathematics and empirical results of this component.

The second step in our pipeline collects the noisy labels from Amazon Mechanical Turk and uses a Bayesian corroboration model to jointly learn the reliability of AMT workers and the true labels of the tweets. Section 4 describes this component. The next section presents our notation and classification model.

2 Notation and Classification Model

We represent each tweet as a feature vector $\mathbf{x}$. We assume these tweet features are generated from a non-stationary distribution since tweets are influenced by external events like trending topics, upcoming celebrities and product releases. We train a binary classifier for each category out of a fixed set: e.g. is this tweet about technology or not? For the purpose of this paper, we will restrict ourselves to one class and denote the class label for $\mathbf{x}$ with $y \in \{-1, +1\}$. Assuming that $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots\}$ is the full history of data points and their categories, our task is to accurately predict $p(y^*|\mathbf{x}^*, \mathcal{D})$, where $\mathbf{x}^*$ and y^* denote a new data point and its corresponding label.

The Twitter firehose generates more than 90 million data points every day. Initially, we have no labels for any of these data points but we have the option of sending any incoming tweet to AMT for human labeling. However, in our experiments each human label costs roughly 0.01$, hence building the full history of labeled data points $\mathcal{D}$ is impossible. Our solution is to obtain human labels for a subset of the history $\mathcal{D}' \subseteq \mathcal{D}$ in a cost-effective, online fashion, such that we can predict

$$p(y^*|\mathbf{x}^*, \mathcal{D}') \approx p(y^*|\mathbf{x}^*, \mathcal{D}).$$

We assume that data points $\mathbf{x} \in \{0, 1\}^N$ are sparse binary vectors, where N is typically very large. Each element x_n represents the presence or absence of certain features, for example, the n'th feature would represent the presence or absence of the word "vuvuzela" in the tweet if this is the $n$the word of the vocabulary. Most features are absent; only a small number are present. As our system is regularly exposed to previously unseen features (e.g. the first occurrence of "vuvuzela" on Twitter), the dimensionality of our feature vectors N increases over time.

The underlying classification model is a linear probit regression model, with $\boldsymbol{\theta}$ being an N-dimensional Gaussian weight vector. Let a latent variable f depend on a linear combination of θ_n's, evaluated only over the *seen* features through $\mathbf{x}^T\boldsymbol{\theta}$. More specifically,

$$p(y, f, \boldsymbol{\theta}|\mathbf{x}, \mathcal{D}') = p(y|f)\, p(f|\mathbf{x}, \boldsymbol{\theta})\, p(\boldsymbol{\theta}|\mathcal{D}') = \Phi(yf)\, \mathcal{N}(f\,;\, \mathbf{x}^T\boldsymbol{\theta}, \sigma^2)\, p(\boldsymbol{\theta}|\mathcal{D}')\ ,$$

where $\Phi(\cdot)$ is the cumulative zero mean, unit variance Gaussian function. The posterior $p(\boldsymbol{\theta}|\mathcal{D}')$ is not analytically tractable, and we replace it with a factorized approximation

$$p(\boldsymbol{\theta}|\mathcal{D}') \approx q(\boldsymbol{\theta}|\mathcal{D}') = \prod_n \mathcal{N}(\theta_n; \mu_n, v_n^2)\ .$$

The parameters of the approximation $q(\boldsymbol{\theta}|\mathcal{D}')$ are determined though Expectation Propagation (EP), or an Assumed Density Filtering (ADF) loop [6]. We obtain $q(\boldsymbol{\theta}|\mathbf{x}, y, \mathcal{D}') = \prod_n \mathcal{N}(\theta_n; m_n, s_n^2)$ by including a new data point $\mathbf{x}, y$ into $\mathcal{D}'$. The update will determine the new m_n, s_n^2 from the projection of

$$\int df\ \Phi(yf)\, \mathcal{N}(f\,;\, \mathbf{x}^T\boldsymbol{\theta}, \sigma^2)\, q(\boldsymbol{\theta}|\mathcal{D}') \rightarrow q(\boldsymbol{\theta}|\mathbf{x}, y, \mathcal{D}')$$

onto a factorized N–dimensional Gaussian. The projection retains the means and diagonal variances of the left hand side into the updated approximation on the right hand side, and drops all other statistics. Because of $\mathbf{x}$ being binary, we have $\mu_n = m_n$ and $v_n = s_n$ for all n where $x_n = 0$, and the two approximations will differ *only* for the features present in $\mathbf{x}$ (i.e. indices where $x_n = 1$). The model is described in detail by [7].

3 Online Active Learning

Consider a previously unseen data point (tweet) $\mathbf{x}$, for which we want to determine the correct class label, y. We have the option of using AMT (which is relatively accurate but comes at a cost) or the classifier (which is free but potentially inaccurate). Our decision criterion is whether the information we expect to gain (according to the beliefs of the current model) by labeling $\mathbf{x}$ compensates sufficiently for the cost of obtaining a human label.

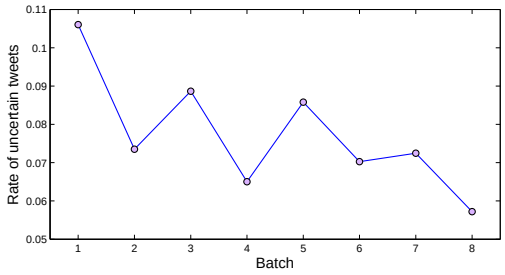

Figure 1: The ratio of tweets for which $\Delta(\mathbf{x}, y)$ is larger than a specified threshold, as a function of model updates after batches submitted to AMT. The batches were of size 2000–4000 Tweets.

We define the *information gain* by the differential entropy between $q(\boldsymbol{\theta}|\mathcal{D}')$ and $q(\boldsymbol{\theta}|\mathbf{x}, y, \mathcal{D}')$. If we add the data point to $\mathcal{D}'$, and the posterior probability mass does not shift at all, then we didn't gain any information. We define the information gain as the Kullback–Leibler divergence,

$$D_{\mathrm{KL}}\Big(q(\boldsymbol{\theta}|\mathcal{D}')\Big\|q(\boldsymbol{\theta}|\mathbf{x}, y, \mathcal{D}')\Big) = \mathbb{E}_{q(\boldsymbol{\theta}|\mathcal{D}')}\left[\log \frac{q(\boldsymbol{\theta}|\mathcal{D}')}{q(\boldsymbol{\theta}|\mathbf{x}, y, \mathcal{D}')}\right] ,$$

where the information gain is with respect to our model q, and not the true posterior $p(\boldsymbol{\theta}|\mathcal{D}')$. We care about the *expected information gain* of learning the true label of $\mathbf{x}$. This expectation should be taken over the true density $p(y|\mathbf{x})$, which is *unknown*. The best that we can do, then, is to use the prediction given by our *current* model, and average over $q(y|\mathbf{x}, \mathcal{D}') = \int d\boldsymbol{\theta}\, df\, p(y|f)\, p(f|\mathbf{x}, \boldsymbol{\theta})\, q(\boldsymbol{\theta}|\mathcal{D}')$. In other words, similar to [5],

$$\Delta(\mathbf{x}, y) = \mathbb{E}_{q(y|\mathbf{x},\mathcal{D}')}\left[\mathbb{E}_{q(\boldsymbol{\theta}|\mathcal{D}')}\left[\log \frac{q(\boldsymbol{\theta}|\mathcal{D}')}{q(\boldsymbol{\theta}|\mathbf{x}, y, \mathcal{D}')}\right]\right] . \tag{1}$$

The inner expectation D_{KL} in (1) can easily be computed for both $y = -1$ and $y = +1$ by making a temporary ADF update to $q(\boldsymbol{\theta}|\mathcal{D}')$, as explained in Section 2. D_{KL} will be a function of only those θ_n that correspond to non-zero entries in $\mathbf{x}$, which will be small compared to N. For the outer sum (expectation) in (1), we determine $q(y = -1|\mathbf{x}, \mathcal{D}')$ and $q(y = 1|\mathbf{x}, \mathcal{D}')$ from the current model.

Figure 1 illustrates the ratio of tweets for which labels are needed, as a function of model updates after batches submitted to AMT.

4 Corroborating User Feedback

The method in the previous section is designed to choose a datapoint that optimally improves the classifier model, under the assumption that AMT users can reliably classify tweets. Unfortunately, we frequently observe that labels from AMT are incorrect: some users are better than others, and even the reliable users can make mistakes. One solution to this problem is to send each classification task to multiple users and use a corroboration model to extract as much useful information from the aggregated labels, while taking user reliabilities into account [4].

To that end, we augment the classifier from the previous section with a number of new random variables. The resulting graphical model is illustrated in Figure 2. We distinguish the following components: as before for each tweet we have an input vector $\mathbf{x}_i$ and a "true" label y_i which we are trying to learn. Let $\boldsymbol{\theta}$ denote the parameters of the linear Gaussian probit regression model. Each AMT user will provide a label for a set of tweets; we denote the label for tweet i by user j as a_{ij}. Note that we do not observe the true labels y_i but only the noisy labels a_{ij}. Finally, each user has a parameter u_j which describes his reliability. For the purpose of this workshop paper, we limit ourselves to a single real value denoting his overall reliability. In a full version of this work (currently under preparation), we describe a method which learns a reliability which can change across topic domains.

The graphical model in Figure 2 corresponds to a joint probability distribution

$$p(\mathbf{a}, \mathbf{u}, \boldsymbol{\theta}, \mathbf{y}|\{\mathbf{x}_i\}, \alpha, \beta) = p(\boldsymbol{\theta})\left(\prod_{i=1}^{n} p(y_i|\mathbf{x}_i, \boldsymbol{\theta}) \prod_{j=1}^{m} p(a_{ij}|y_i, u_j)\right)\left(\prod_{j=1}^{m} p(u_j)\right) \tag{2}$$

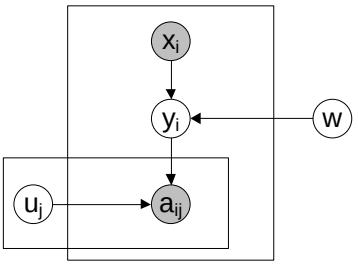

Figure 2: The graphical model for the Bayesian corroboration model.

We use a linear gaussian probit likelihood for $p(y_i|x_i, w)$ with a Gaussian prior on $p(w)$ as described in the previous section. We use a $u_j \sim \text{Beta}(\alpha, \beta)$ prior over the user reliabilities and a conditional probability for $p(a_{ij}|y_i, u_j)$ illustrated in Table 1 below. The interpretation of this condition distribution is that for a true fact, with probability u_j a user reports the true value and with probability $1 - u_j$ the user reports the wrong value. This is a Bayesian version of the work in [3].

a_{ij} \ y_j	T	F
T	u_j	$1 - u_j$
F	$1 - u_j$	u_j

Table 1: The conditional probability table $p(a_{ij}|y_j, u_i)$.

5 Discussion

Based on the case of online classification for tweets, we illustrated a general approach for cost-efficient online classification. Our model combines the active learning paradigm with a model for corroborating user opinions. In a large-scale system such as Project Emporia, which is daily exposed to tens of millions of tweets, cost-efficiency is a critical issue. The presented classification model has been tested on real-world scenarios and preliminary experimental results indicate its viability.

We are currently preparing an extended version of this paper, where we describe our experimental results in more detail and explore further extensions to the corroboration model to capture more fine-grained user expertise and reliabilities. Finally, we are also looking into decision-theoretic formalism which based on the Bayesian corroboration makes a decision whether to accept a completed task from Amazon Mechanical Turk or not.

References

[1] FUSE Labs `http://fuse.microsoft.com/`

[2] Project Emporia `http://www.projectemporia.com/`

[3] Dawid, A.P., Skene, A.M.: Maximum likelihood estimation of observer error-rates using the EM algorithm In Applied Statistics, 28, 1, 20–28, 1979

[4] Kasneci, G., Van Gael, J., Herbrich, R., Graepel, T.: Bayesian Knowledge Corroboration with Logical Rules and User Feedback. In: ECML PKDD 2010, Springer 2010

[5] MacKay, D.J.C.: Information-based objective functions for active data selection. In Neural Computation, 4, 4, 589–603, 1992

[6] Minka, T.P.: A family of algorithms for approximate Bayesian inference. Ph.D. thesis, Massachusetts Institute of Technology. 2001

[7] Graepel, T., Quinonero Candela, J., Borchert, T., and Herbrich, R.: Web-Scale Bayesian Click-Through Rate Prediction for Sponsored Search Advertising in Microsoft's Bing Search Engine, In Proceedings of the 27th International Conference on Machine Learning ICML 2010, June 2010