Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Finding One's Best Crowd: Online Learning by Exploiting Source Similarity

Yang Liu and Mingyan Liu
Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor
1301 Beal Avenue, Ann Arbor, Michigan 48109
{youngliu,mingyan}@umich.edu

Abstract

We consider an online learning problem (classification or prediction) involving disparate sources of sequentially arriving data, whereby a user over time learns the best set of data sources to use in constructing the classifier by exploiting their similarity. We first show that, when (1) the similarity information among data sources is known, and (2) data from different sources can be acquired without cost, then a judicious selection of data from different sources can effectively enlarge the training sample size compared to using a single data source, thereby improving the rate and performance of learning; this is achieved by bounding the classification error of the resulting classifier. We then relax assumption (1) and characterize the loss in learning performance when the similarity information must also be acquired through repeated sampling. We further relax both (1) and (2) and present a cost-efficient algorithm that identifies a *best crowd* from a potentially large set of data sources in terms of both classifier performance and data acquisition cost. This problem has various applications, including online prediction systems with time series data of various forms, such as financial markets, advertisement and network measurement.

Introduction

The ability to learn (classify or predict) accurately with sequentially arriving data has many applications. Examples include predicting future values on a prediction market, weather forecasting, TV ratings, and ad placement by observing user behavior. The subject of learning in such contexts has been extensively studied. Past literature is heavily focused on learning by treating each source or object's historical data separately, see e.g., (Lai and Robbins 1985; Lu, Pl, and Pal 2010; Langford and Zhang 2007) for single source multi-armed bandit problems for learning the best options of returned rewards, (Kim 2003) for a support vector machine based forecasting for financial time series data, (Hao et al. 2009) for a model predicting spammers using a network's past statistics, and (Hyup Roh 2007) for forecasting stock price index, among other.

More recent development has increasingly been focusing on improving learning through integrating data from multiple sources with similar statistics, see e.g., (He et al.) for wind power prediction using both temporal and spatial information. The idea of increasing sample spaces by exploiting similarity proves to be helpful especially when the data arrives slowly, e.g., weather reports generated a few times per day. This idea naturally arises when different data sources are physically correlated, e.g., wind turbines on the same farm, or environmental monitoring sensors located within close proximity. However, it also fits well in the emerging context of crowdsourcing, where different sources (e.g., Amazon Mechanical Turks) contribute to a common data collection objective (e.g., labeling a set of images), and exploiting multiple data sources can improve the quality of crowdsourced data. For instance the idea of aggregating selectively data from a crowd to make prediction more accurate is empirically demonstrated and referred to as finding a "smaller but smarter crowd" in (Galesic and Barkoczi 2014; Goldstein, McAfee, and Suri 2014).

In this paper we seek to make the notion of a "smarter" crowd quantitatively precise and develop methods to systematically identify and utilize this crowd. Specifically, we consider a problem involving K (potentially-)disparate data sources, each of which may be associated with a user. A given user can use its own data to achieve a certain learning (prediction, classification) objective but is interested in improving its performance by tapping into other data sources, and can request data from other sources at a cost. Accordingly, decisions need to be made judiciously on which sources of data should be used so as to optimize its learning accuracy. This implies two challenges: (1) we need to be able to measure the similarity/disparity between two sources in order to differentiate which sources are more useful toward the learning objective, and (2) we need to be able to determine the best set of sources given the measured similarity. Prior work most relevant to the present study is (Crammer, Kearns, and Wortman 2008), where the problem of combining static IID data sources is analyzed. There are however a number of key differences: 1) in (Crammer, Kearns, and Wortman 2008) the similarity information is assumed known a priori and the cost of obtaining data is not considered. 2) The results in (Crammer, Kearns, and Wortman 2008) are established pre-collected IID data, while we focus on an online learning setting with Markovian data sources. In addition, the methodology we employ in this paper is quite different from (Crammer, Kearns, and Wortman 2008)

which draws mainly from VC theory (Vapnik 1995), while our study is based on both VC theory and the multi-armed bandit (MAB) literature (Auer, Cesa-Bianchi, and Fischer 2002).

We will start by establishing bounds on the expected learning error under ideal conditions, including that (1) the similarity information between data sources is known a priori, and (2) data from all sources are available for free. We then relax assumption (1) and similarly establish the bounds on the error when such similarity information needs to be learned over time. We then relax both (1) and (2) and design an efficient online learning algorithm that simultaneously makes decisions on requesting and combining data for the purpose of training the predictor, and learning the similarity among data sources. We again show that this algorithm achieves a guaranteed performance uniform in time, and the additional cost with respect to the minimum cost required to achieve optimal learning rate diminishes in time. Moreover, the obtained bounds show clearly the trade-off between learning accuracy and the cost to obtain additional data. This provides useful information for system designers with different objectives. To our best knowledge this is the first study on online learning by exploiting source similarity with provable performance guarantees. Unless otherwise specified, all proofs can be found in (Liu and Liu 2015).

Problem Formulation

Learning with multiple data sources

Consider K sources of data each associated with a unique user, indexed by $\mathcal{D} = \{1,2,\cdots,K\}$, which we also refer to as the whole crowd of sources. The sources need not be governed by identical probability distributions. Data samples arrive in discrete time to each user; the sample arriving at time t for user i is denoted by $z_i(t) = (x_i(t), y_i(t))$, $t = 1,2,\cdots$, with $x_i(t)$ denoting the features and $y_i(t)$ denoting the labels. At each time t, $x_i(t)$ is revealed first followed by a prediction on $y_i(t)$ made by the user, after which $y_i(t)$ is revealed and $z_i(t)$ is added to the training set. For simplicity of exposition, we will assume $x_i(t)$ to be a scalar; however our analysis easily extends to more complex forms of data, including batch arrivals. The objective of each user is to train a classifier to predict $y_i(t)$ using collected past data, and after prediction at time t, $y_i(t)$ will be revealed and can be used for training in the future steps. As a special case, when the target is to predict for future, $y_i(t)$ can be taken as $x_i(t+1)$. For analytical tractability we will further assume that the data arrival processes $\{x_i(t)\}_t$, $\forall i$, are mutually independently (but not necessarily identical), and each is given by a first order[1] finite-state positive recurrent Markov chain, with the corresponding transition probability matrix denoted by P^i on the state space $\mathcal{X}^i$ ($|\mathcal{X}^i| < \infty$). Denote by $P^i_{x,y}$ the transition probability from state x to y under P^i, and by π^i its stationary distribution on $\mathcal{X}^i$. For simplicity we will assume that $\mathcal{X}^1 = \mathcal{X}^2 = ... = \mathcal{X}^K = \mathcal{X}$, though this assumption can be easily relaxed, albeit with more cumbersome notation. The motivation for such modeling choice is by observing that for many applications the sequentially arriving data does not follow IID distribution as has been studied in the literature; consider e.g., weather conditions. Suppose labels $y_i(t) \in \mathcal{Y}^i$ and again for simplicity let us assume $\mathcal{Y}^1 = \mathcal{Y}^2 = ... = \mathcal{Y}^K = \mathcal{Y}$, and $|\mathcal{Y}| < \infty$. Denote $y^* := \max_{y\in\mathcal{Y}} |y|$.

For the classification job, a straightforward approach would be for each user i to build a classifier/predictor by using past observations of its own data up to time t: $\{z_i(1),...,z_i(t)\}$. Denote the classifier by f_i for user i, and a loss function $\mathcal{L}$ to measure the classification error. For instance $\mathcal{L}$ can be taken as the squared loss function $\mathcal{L}(f_i, z_i(t)) = [y_i(t) - f_i(x_i(t))]^2$. With the definition of loss function, the classification task for a user is to find the classifier that best fits its past observations:

$$f_i(t) = \text{argmin}_{f\in\mathcal{F}} \sum_{n=1}^{t} \mathcal{L}(f, z_i(n)) \,, \tag{1}$$

where we have used $\mathcal{F}$ to denote the set of all models of classifier (hypothesis space). For example, $\mathcal{F}$ could contain the classical linear regression models.

The idea we seek to explore in this paper is to construct the classifier f_i by utilizing similarity embedded among data sources, i.e., we ask whether f_i should be a function of all sources' past data and not just i's own, and if so how should such a classifier be constructed. Specifically, if we collect data from a set Ω_k of sources and use them as if they were from a single source, then the best classifier is given by

$$f_{\Omega_k}(t) = \text{argmin}_{f\in\mathcal{F}} \sum_{j\in\Omega_k} \sum_{n=1}^{t} \mathcal{L}(f, z_j(n)) \,. \tag{2}$$

It was shown in (Crammer, Kearns, and Wortman 2008) that the expected error of the above classifier is bounded by a function of certain source similarity measures; the higher the similarity the lower the error bound.

Our interest is in constructing the best classifier for any given user i by utilizing other data sources. To do so we will need to measure the similarity or discrepancy between sources and to judiciously use data from the right set of sources. We will accomplish this by decomposing the problem into two sub-problems, the first is to use a similarity measure to determine a preferred set Ω_k^* to use, and the second is to construct the classifier using data from this set.

Pair-wise similarity between data sources

We first introduce the notion of cross-classification error, which is the expected loss when using classifier f_j (trained using source j's data) on user i's data and can be formally defined as $r_i(f_j) = E_i[\mathcal{L}(f_j, z_i)]$ where the expectation is with respect to user i's source data distribution. In principle, this could be used to measure the degree of similarity between two data sources i and j. However, this definition is not easy to work with as it involves a classifier that is only implicitly given in (1). Instead, we introduce a notion of similarity between two data sources i and j, that satisfies the following two conditions: (1) it can be obtained from the statistics of two respective data sources, and (2) it satisfies

[1] A high order extension is also straightforward.

the following bound:

$$r_i(f_j) \leq \beta_1(1-S_{i,j})+\beta_2 \ , \tag{3}$$

where $\beta_1, \beta_2 \geq 0$ are normalization constants and $0 \leq S_{i,j} \leq 1$ denotes the similarity measure; the higher this value the more similar two sources. The relationship captured in Eqn (3) between the error function and similarity can also take on alternate forms; we adopt this simple linear relationship for simplicity of exposition. The following example shows the existence of such a measure.

Suppose for each user i, corresponding to each state/feature $x \in \mathcal{X}$, labels $y \in \mathcal{Y}$ is generated according a probability measure Q^i_x and denote each probability as $Q^i_{x,y}$ and $\sum_{y\in\mathcal{Y}} Q^i_{x,y} = 1$. Consider the following example. Take $\mathcal{L}$ as the squared loss and $S_{i,j}$ as:

$$S_{i,j} = 1 - \max_{x\in\mathcal{X}, y\in\mathcal{Y}} |Q^i_{x,y} - Q^j_{x,y}|^2 \ . \tag{4}$$

Then we can show[2] that, by setting $\beta_1 := 2\sum_{y\in\mathcal{Y}} y^2$ and $\beta_2 := 2\sum_{x\in\mathcal{X}} \pi^i_x \cdot \sum_{y\in\mathcal{Y}} Q^i_{x,y} \cdot (\sum_{\hat{y}\in\mathcal{Y}} Q^i_{x,\hat{y}}\hat{y} - y)^2$, i.e. two times the intrinsic classification error with user i's own (perfect) data, which is independent with other sources j, the choice of $S_{i,j}$ satisfies both conditions. We note that the choice of such an S is not unique. For example, we could also take $S_{i,j}$ to be

$$S_{i,j} = 1 - \sum_{x\in\mathcal{X}} \pi^i_x \cdot \sum_{y\in\mathcal{Y}} |Q^i_{x,y} - Q^j_{x,y}|^2 \ ,$$

while setting $\beta_1 := 2(y^*)^2$. Later we will argue that an S that leads to a tighter bound can help achieve a better performance in classification. As it shall become clearer later when such similarity information needs to be estimated, the tradeoff between selecting a tighter and looser similarity measure comes from the fact that tighter similarity may incur more learning error as it requires the evaluation of more terms.

Without loss of generality, for the remainder of our discussion we will focus on user 1. We will also denote $s_i := \min\{S_{1,i}, S_{i,1}\}, \forall i$. While the definition given in (4) is symmetric in i and j such that $S_{1,i} = S_{i,1}$, this needs not be true in general under alternate definitions of similarity. Note that $s_1 = 1$. We will then relabel the users in decreasing order of their similarity to user 1: $1 = s_1 \geq s_2 .... \geq s_K \geq 0$.

Solution with Complete Information

As mentioned earlier, the problem of finding the best set of data sources to use and that of finding the best classifier given this set are inherently coupled and strictly speaking need to be jointly optimized, resulting in significant challenges. The approach we take in this paper is as follows. We will first derive an upper bound on the error of the classifier given in (2) when applied to user 1, for a set of k independent Markov sources; this bound is shown to be a function of k and their similarity with user 1. This bound is then optimized to obtain the best set. Below we derive this upper bound assuming (1) the similarity information is known and (2) data is free, i.e., at time t all past and present samples from all sources are available to user 1.

[2]Please refer to Appendices.

Upper bounding the learning error

First notice we have the following convergence results for positive recurrent Markov Chain we consider in the current paper (Rosenthal 1995),

$$||\tilde{\pi}^i(t) - \pi^i||_{\text{TV}} \leq C_{\text{MC}} \cdot (\lambda^i_2)^t \ ,$$

where C_{MC} is some positive constant, $\tilde{\pi}^i_x(t)$ is the expected empirical distribution of state x for data source i's Markov chain upto time t for user i and π^i_x denotes its stationary distribution, and $0 < \lambda^i_2 < 1$ is the second largest eigenvalue which specifies the mixing speed of the process. The total variation distance $||p-q||_{\text{TV}}$ between two probability measures p and q that are defined on $\mathcal{X}$ is defined as follows

$$||p-q||_{\text{TV}} := \max_{S\in 2^{\mathcal{X}}} |\sum_{x\in S} (p(x)-q(x))| \ . \tag{5}$$

Denote $\rho_{k(t)}(t) := \max\mathcal{L} \cdot C_{\text{MC}} \frac{\sum_{i\in k(t)} (\lambda^i_2)^t}{|k(t)|}$, where $\max\mathcal{L}$ is the maximum value attained by the loss function. Throughout the paper we denote $[k] := \{1, 2, ..., k\}$ as the ordered and continuous set up to k, and $k(t)$ for any other un-ordered set invoked at time t and use $|k(t)|$ to denote its size. For squared loss function we have the following results:

Theorem 1. *At time t, with probability at least $1 - O(\frac{1}{t^2})$ the error of a classifier $f_{[k]}(t)$ constructed using data from k sources of similarity $s_i, i \in k(t)$ can be bounded as*

$$r_1(f_{k(t)}(t)) \leq \underbrace{4\min_{f\in\mathcal{F}} r^{\text{IID}}_1(f) + 6\beta_2}_{\text{Term 1}} + \underbrace{6\beta_1 \frac{\sum_{i\in k(t)}(1-s_i)}{|k(t)|}}_{\text{Term 2}}$$
$$+ \underbrace{\rho_{k(t)}(t)}_{\text{Term 3}} + \underbrace{8y^*(2\sqrt{2d}+y^*)\sqrt{\frac{\log|k(t)|t}{|k(t)|t}}}_{\text{Term 4}} \ , \tag{6}$$

where d is the VC dimension for $\mathcal{F}$, and $r^{\text{IID}}_1(f)$ is the expected prediction error when the data are generated according to an IID process.

Denote the upper bound for $r_1(\cdot)$ in Eqn. (6) with set $k(t)$ of data sources (after ordering based on their similarity with user 1) at time t by $\mathcal{U}_{k(t)}(t)$. The results may be viewed as an extension to the previous one from (Crammer, Kearns, and Wortman 2008) where static and IID data sources were considered. This upper bound can serve as a good guide for the selection of such a set and in particular the best choice of $|k(t)|$ given estimated values of s_i's. Note that Terms 1 is independent of this selection and it is a function of the baseline error of the classification problem, Term 2 is due to the integration of disparate data sources, Term 3 comes from the mixing time of a Markov source, and Term 4 arises from imperfect estimation and decision using a finite number of samples ($|k(t)|t$ samples up to time t).

Below we first point out the key steps in the proof that differ from that in (Crammer, Kearns, and Wortman 2008) (full proof is in the supplementary materials), and then highlight the properties of this bound.

Main steps in the proof Our analysis starts with connecting Markovian data sources to IID sources so that the classical VC theory (Vapnik 1995) and corresponding results can apply. The idea is rather simple: by the ergodicity assumption on the arrival process, the estimation error converges to that of IID data sources as shown in (Adams and Nobel 2010). In particular, we can bound the difference in error when applying a predictor $f \in \mathcal{F}$ to a Markovian vs. an IID source (with distribution being the same as the steady state distribution of the Markov chain) at time t, constructed with available data as follows:

$$\begin{aligned}&|r_i(f(t)) - r_i^{\text{IID}}(f(t))| \\ &= |\sum_{x\in\mathcal{X}} \tilde{\pi}_x^i E_{y\sim\mathcal{Y}}[\mathcal{L}(f(t),(x,y))] - \sum_{x\in\mathcal{X}} \pi_x^i E_{y\sim\mathcal{Y}}[\mathcal{L}(f(t),(x,y))]| \\ &\le \max \mathcal{L} \cdot C_{\text{MC}}(\lambda_2^i)^t \ .\end{aligned}$$

We impose α-triangle inequality on the error function $\forall i,j,k$, of the corresponding data sources $r_i(f_j) \le \alpha \cdot [r_i(f_k) + r_k(f_j)]$, where $\alpha \ge 1$ is a constant. When $\mathcal{L}$ is the squared loss function, we have $\alpha = 2$, following Jensen's inequality. Then $\forall f$

$$\frac{r_1(f)}{k} \le \frac{\alpha \cdot [r_1(f_i) + r_i(f)]}{k} \ .$$

Sum over all $i \in k(t)$ we have

$$r_1(f) \le \frac{\alpha\beta_1}{|k(t)|} \cdot \sum_{i\in k(t)} (1-s_i) + \alpha\beta_2 + \alpha \cdot \bar{r}_{k(t)}(f) \ ,$$

where $\bar{r}_{k(t)}(f) = \frac{\sum_{i\in k(t)} r_i(f)}{|k(t)|}$ is the average regret by applying f onto the $|k(t)|$ data sources. Due to the bias of mixing time for Markovian sources we have the following fact :

$$\bar{r}_{k(t)}(f) \le \bar{r}_{k(t)}^{\text{IID}}(f) + \rho_{k(t)}(t).$$

The rest of the proof focuses on bounding $\bar{r}_{k(t)}^{\text{IID}}(f)$, i.e., the expected prediction error on IID data sources, which is similar in spirit to that presented in (Crammer, Kearns, and Wortman 2008).

Properties of the error bound The upper bound $\mathcal{U}_{k(t)}(t)$ has the following useful properties.

Proposition 2. *For sources ordered in decreasing similarity $s_1 \ge s_2 \cdots$, $\frac{\sum_{i=1}^k s_i}{k}$ is non-increasing in k.*

This is straightforward to see by noting that

$$\frac{\sum_{i=1}^{k+1} s_i}{k+1} - \frac{\sum_{i=1}^k s_i}{k} = -\sum_{i=1}^k \frac{s_i}{k(k+1)} + \frac{s_{k+1}}{k+1} = \sum_{i=1}^k \frac{s_{k+1} - s_i}{k(k+1)} \le 0 \ .$$

Terms 3 and 4 both decrease in time. While Term 4 converges at the order of $O(1/\sqrt{t})$, Term 3 converges with geometric rate, which is much faster than Term 4 and can be ignored for now. We then know because of the use of multiple sources, Term 4 decrease $|k(t)|$ times faster, leading to a better bound. This shows how the use of multiple sources fundamentally changes the behavior of the error bound.

The upper bound also suggests that the optimal selection is always to choose those with the highest similarity, which leads to a linear search for the optimal number k. Based on above discussions, the trade-off comes from the fact a larger k returns a smaller average similarity term $\sum_{i=1}^k s_i/k$ (and thus a larger $\sum_{i=1}^k (1-s_i)/k$), while with more data we have a faster convergence of Term 4. Define the optimal set of sources at time t as the one minimizing the bound $\mathcal{U}_{k(t)}(t)$, and denote it by $k^*(t)$. We then have the following fact,

Proposition 3. *When $\{s_i\}_{i\in\mathcal{D}}$ is known, $\exists\, t_o$, such that $\forall t \ge t_o$, if $i \in k^*(t)$ then $i \in k^*(n), \forall t_o \le n \le t$.*

The proof can be found in Appendices. This implies that if a data source is similar enough to be included at t, then it would have been included in previous time steps as well except for a constant number of times. This also motivates us to observe a threshold or phase transitioning phenomenon in selecting each user's best crowd. This result is also crucial in proving Theorem 6 where it helps establish bounded number of missed sampling for an optimal data source in an adaptive algorithm.

Proposition 4. *A set of tighter similarity measures S returns better worst case performance.*

Consider two such similarity measures s' and s with $s'_i \ge s_i$ (with at least one strict inequality). Suppose at any time t and optimal set of crowd for s is $k(t)$, then simply by selecting $k(t)$ for s' we achieve a better worst case performance (a smaller $\sum_{i=1}^k (1-s_i)/k$ in upper bound).

Overhead of Learning Similarity

As we show in the previous section, once the optimal set of data sources is determined, the classification/prediction performance is bounded. However in a real crowdsourcing system, neither of the two assumptions may be valid. In this section we relax the first assumption and consider a more realistic setting where the similarity information remains unknown a-priori and can only be learned through shared data. In this regards we need to estimate the similarity information $\{s_i\}_{i\ne 1}$ while making decision of which set of data sources to use.

The learning process works in the following way. At step t, we first estimate similarity $\tilde{s}_i$ according to the following:

$$\tilde{s}_i = 1 - \max_{x\in\mathcal{X}, y\in\mathcal{Y}} |\tilde{Q}_{x,y}^i(t) - \tilde{Q}_{x,y}^1(t)|^2 \ ,$$

where $\tilde{Q}_{x,y}^i(t) := \frac{n_{i,x\to y}(t)}{n_{i,x}(t)}$ are the estimated transition probability matrices with $n_{i,x}(t)$ denoting the number of times user i is sampled to be in state $x \in \mathcal{X}$ up to time t and $n_{i,x\to y}(t)$ denoting the number of observed samples from data source i being in (x,y). Different from the previous Section, now since $\{s_i\}_{i\ne 1}$ is unknown, in order to select data sources, the estimate of the upper bound $\mathcal{U}_{k(t)}(t)$ becomes a function of $\{\tilde{s}_i\}$: $\mathcal{U}_{k(t)}(t;\{\tilde{s}_i\}_{i\in k(t)})$, which is obtained by simply substituting all s terms in $\mathcal{U}_{k(t)}(t)$ with $\tilde{s}$. Denote the terms that are being affected by choosing set $k(t)$ in $\mathcal{U}_{k(t)}(t;\{\tilde{s}_i\}_{i\in k(t)})$ as follows:

$$\tilde{\mathcal{U}}_{k(t)}^{tr}(t) = 6\beta_1 \frac{\sum_{i\in k(t)}(1-\tilde{s}_i)}{k} + 8y^*(2\sqrt{2d} + y^*)\sqrt{\frac{\log|k(t)|t}{|k(t)|t}} \ .$$

Note we are omitting $\rho_{k(t)}(t)$ as it is on a much smaller order and will not affect our results order-wise.

Then the learning algorithm first orders all data sources according to $\{\tilde{s}_i\}$. And then choses $\tilde{k}^*(t)$ by a linear search such that

$$\tilde{k}^*(t) = \arg \max_{[k],1\le k\le K} \tilde{\mathcal{U}}^{tr}_{[k]}(t) \ .$$

We have the following results.

Theorem 5. *At time t, with probability at least $1-O(\frac{1}{t^2})$ the error of trained classifier $f_{\tilde{k}^*(t)}(t)$ using $\tilde{k}^*(t)$ data sources can be bounded as follows*

$$r_1(f_{\tilde{k}^*(t)}(t)) \le \mathcal{U}_{k^*(t)}(t) + O(\sqrt{\frac{\log t}{t}}) \ . \qquad (7)$$

Clearly from above results we see there is an extra $O(\sqrt{\log t/t})$ term capturing the loss of learning the similarity information.

A Cost-efficient Algorithm

Now we relax the second restriction on data acquisition. In reality data acquisition from other sources are costly. In our study, we explicitly model this aspect whereby at each time step a user may request data from another user at a unit cost of c. This modeling choice not only reflects reality, but also allows us to examine the tradeoff between a user's desire to to keep its overall cost low while keeping its prediction performance high. We present a cost-efficient algorithm with performance guarantee. As one may expect, with less data the prediction accuracy will degrade. But the number of unnecessary data will also be bounded from above.

A cost-efficient online algorithm

Denote by $n_i(t)$ the number of collected samples from source i up to time t and $N_{k(t)}(t) = \sum_{i\in k(t)} n_i(t)$. Notice in this section $n_i(t) \neq t$ in general. Denote $D(t) := O(t^z)$; z will be referred to as the exploration constant satisfying $0 < z < 1$. Later we will show how z controls the trade-off between data acquisition and classification accuracy. Again denote by $n_{i,x}(t)$ the number of times user i is sampled to be in state $x \in \mathcal{X}$ up to time t and construct the following set at each time t :

$$O(t) = \{i : i \in \mathcal{D}, \exists\, x \in \mathcal{X}, n_{i,x}(t) < D(t)\} \ .$$

We name the algorithm as `K-Learning`, which consists mainly of the following two steps (run by user 1):

Exploration: At time t, if any data source has a state x that has been observed (from requested data) for less than $D(t)$ times, i.e., if $O(t)$ is non-empty, then the algorithm enters an exploration phase and collects data from *all* sources $k_2(t) = \mathcal{D}$ and predicts via its own data $k_1(t) = \{1\}$. The prediction at exploration phase is *conservative* since without enough sampling user 1 cannot be confident in calculating its optimal set of similar sources, in which case the user would rather limit itself to its own data.

Exploitation: If $O(t)$ is empty at time t then the algorithm enters an exploitation phase, whereby it first estimates similarity measures of all sources. For our analysis we will use the same definition given earlier: $\tilde{s}_i(t) = 1 - \max_{x\in\mathcal{X},y\in\mathcal{Y}} |\tilde{Q}^i_{x,y}(t) - \tilde{Q}^1_{x,y}(t)|^2$. The algorithm then calculates $k_1(t)$ using the estimated bound $\tilde{\mathcal{U}}^{tr}_{k_1(t)}(t)$, and uses data from this set $k_1(t)$ of sources for training the classifier, while requesting data from set $k_2(t)$, where $k_2(t)$ is set to be:

$$k_2(t) := \text{argmax}_{k'(t)\subseteq D}\{|k'(t)| : \tilde{\mathcal{U}}^{tr}_{k'(t)}(t) \in [\tilde{\mathcal{U}}^{tr}_{k_1(t)}(t) - \sqrt{\frac{\log t}{t^z}}, \tilde{\mathcal{U}}^{tr}_{k_1(t)}(t) + \sqrt{\frac{\log t}{t^z}}]\} \ .$$

Notice when calculating $k_2(t)$ we set a tolerance region (due to imperfect estimation of $\tilde{\mathcal{U}}^{tr}_{k_1(t)}(t)$) so that a sample data from an optimal data source will not be missed with high probability.

Algorithm 1 `K-Learning`

Initialization:
Set $t = 1$ and similarity $\{\tilde{s}_i(1)\}_{i\in\mathcal{D}}$ to some value in $[0,1]$; $n_{i,x}(t) = 1$ for all i and x.
loop:
Calculate $O(t)$.
if $O(t) \neq \emptyset$ **then**
    Explores, sets $k_1(t) = \{1\}, k_2(t) = \mathcal{D}$.
else
    Exploit, orders data sources according to $\{\tilde{s}_i(t)\}_{i\in\mathcal{D}}$ and computes $k_1(t)$ that minimizes $\tilde{\mathcal{U}}^{tr}_{k_1(t)}(t)$, which is solved using the linear search property, and the current estimates $\{\tilde{s}_i(t)\}_{i\in\mathcal{D}}$. Set $k_2(t)$ as $k_2(t) := \text{argmax}_{k'(t)\subseteq D}\{|k'(t)| : \tilde{\mathcal{U}}^{tr}_{k'(t)}(t) \in [\tilde{\mathcal{U}}^{tr}_{k_1(t)}(t) - \sqrt{\frac{\log t}{t^z}}, \tilde{\mathcal{U}}^{tr}_{k_1(t)}(t) + \sqrt{\frac{\log t}{t^z}}]\}$.
end if
Construct classifier $f_{k_1(t)}$ using data collected from sources in $k_1(t)$. Request data from $k_2(t)$.
$t := t+1$ and update $\{n_{i,x}(t)\}_{i,x}$, $\{\tilde{s}_i(t)\}_{i\in\mathcal{D}}$ using collected samples.
goto *loop*.

Performance of `K-Learning`

There are three types of error in the learning performance: (1) Error due to exploration, in which case the error comes from conservative training due to no enough sampling. Due to technical difficulties, we approximate the error (compared to the performance with optimal classifier) by the worst case performance loss, that is the performance difference in upper bounds. (2) Prediction error associated with incorrect computation of $k_1(t)$ (i.e., $k_1(t) \neq k^*(t)$) in exploitation due to imperfect estimates on $\{s_i\}_{i\neq 1}$. (3) Prediction error from sub-sampling effects. This is because even though under the case that $k_1(t) = k^*(t)$, i.e., $k^*(t)$ is correctly identified, due to incomplete sampling, $\exists\ i > 1, n_i(t) < t$, $\hat{\mathcal{U}}_{k_1(t)} \neq \mathcal{U}_{k_1(t)}$, where $\hat{\mathcal{U}}_{k_1(t)}$ is the upper bound for the classification error with collected data: this can be similarly derived following the proof of Theorem 1 and results in (Crammer, Kearns,

and Wortman 2008):

$$\hat{\mathcal{U}}_{k(t)}(t) = 4\min_{f\in\mathcal{F}} r_1^{\widetilde{\text{IID}}}(f) + 6\beta_2 + 6\beta_1 \frac{\sum_{i\in k(t)} n_i(t)(1-s_i)}{N_{k(t)}(t)} + \tilde{\rho}_{k(t)}(t) + 8y^*(2\sqrt{2d}+y^*)\cdot\sqrt{\frac{\log N_{k(t)}(t)}{N_{k(t)}(t)}} ,$$

where $\tilde{\rho}_{k(t)} := \max \mathcal{L}\cdot C_{\text{MC}} \frac{\sum_{i\in k(t)}(\lambda_2^i)^{n_i(t)}}{|k(t)|}$, and $\min_{f\in\mathcal{F}} r_1^{\widetilde{\text{IID}}}(f)$ is error rate over a biased data distribution due to incomplete sampling, compared to the target IID distribution. We emphasize that the difference between $\mathcal{U}_{k(t)}(t;\{\tilde{s}_i\}_{i\in k(t)})$ and $\hat{\mathcal{U}}_{k(t)}(t)$: $\mathcal{U}_{k(t)}(t;\{\tilde{s}_i\}_{i\in k(t)})$ is the estimation of upper bound $\mathcal{U}_{k(t)}(t)$ with estimated similarity information $\tilde{s}$, while $\hat{\mathcal{U}}_{k(t)}(t)$ bounds actual error of the learning task at each step. In $\mathcal{U}_{k(t)}(t)$ and $\mathcal{U}_{k(t)}(t;\{\tilde{s}_i\}_{i\in k(t)})$, full samples are assumed to have been collected for each data source in $k(t)$, i.e., $n_i(t) = t$. However this is not true for $\hat{\mathcal{U}}_{k(t)}(t)$, except for $n_1(t)$ the data source for user 1 itself. Also due to discontinuous sampling for Markovian data, the sampled data distribution is biased which results in $\min_{f\in\mathcal{F}} r_1^{\widetilde{\text{IID}}}(f)$. The main gist of bounding this discrepancy is that due to Proposition 3 we are able to bound the missed samples for a data source appearing in the optimal set.

A subtle difference between the results in this section and the previous one is the performance of the classifier trained during an *exploration* phase is simply the one using user 1's own data, which is bounded away from the optimal performance bound (via data sources $k^*(t)$). Denote the worse case performance loss (difference in performance upper bound) in exploration phases upto time t as $R_e(t)$, that is

$$R_e(t) = \sum_{n=1}^{t} \mathbf{1}_{O(n)\neq\emptyset}\cdot|\mathcal{U}_{[1]}(t) - \mathcal{U}_{k^*(t)}(t)| . \tag{8}$$

This is a quantity we are interested in determining for exploration phases. For exploitation phases, we evaluate the prediction/classification performance as the ones with classifier $f_{k_1(t)}(t)$.

Theorem 6. *At time t,*

- *The number of exploration phases is bounded as follows,*

$$E[\sum_{n=1}^{t} \mathbf{1}_{O(n)\neq\emptyset}] \leq O(t^z) .$$

Further the per round performance loss due to exploration phases $\frac{E[R_e(t)]}{t}$ is bounded as follows: with probability being at least $1-O(e^{-Ct^z})$ where $C>0$ is a constant,

$$\frac{E[R_e(t)]}{t} \leq O(\sqrt{z\cdot\log t}\cdot t^{z/2-1}) .$$

- *If t is an exploitation phase, with probability being at least $1-O(\frac{1}{t^2})$ we bound the average prediction error for classifier $f_{k_1(t)}(t)$ with data sources $k_1(t)$ as follows,*

$$r_1(f_{k_1(t)}(t)) \leq \mathcal{U}_{k^*(t)}(t) + O(\sqrt{\frac{\log t}{t^z}}) + O(\log t\cdot t^{-2/3}) .$$

Note on the bound:

- $O(\sqrt{z\cdot\log t}\cdot t^{z/2-1})$ is the average error invoked by exploration. This term is diminishing with t, that is the average amount of exploration error is converging to 0. $O(\sqrt{\log t/t^z})$ is the learning error incurred in exploitation phases, which is in analogy to the $O(\sqrt{\log t/t})$ term as shown in the bound proved in Theorem 5. $O(\log t\cdot t^{-2/3})$ is also incurred in exploitation phases. This is a unique error associated with subsampling of Markovian data: due to (1) missed sampling and (2) discontinuous sampling.
- It should be noted that the prediction error term $O(\sqrt{\log t/t^z})$ decrease with z for $0<z<1$. That is with a higher z, a tighter bound can be achieved. With $z\to 1$ (number of samples cannot go beyond t at time t), we can show the prediction error term converges to $O(\sqrt{\log t/t})$, which is consistent with the results we reported in last section. Also it worths pointing out $O(\log t\cdot t^{-2/3})$ is generally on a smaller order compared to $O(\sqrt{z\cdot\log t}\cdot t^{z/2-1})$ and $O(\sqrt{\log t/t^z})$: simply set z to be $z>2/3$.
- This observation also sheds lights on establishing the tightness of this bound for z close to 1, as $O(\sqrt{\log t/t})$ is the uniform convergence bound as proved in statistical learning theory (Vapnik 1995).

Cost analysis

To capture the effectiveness of cost saving, we define the following difference in cost:

$$\text{Cost measure} : R_c(t) = c\sum_{n=1}^{t}\sum_{i=1}^{K} \mathbf{1}_{i\notin k^*(n), i\in k_2(n)} .$$

$R_c(t)$ will be referred to as the cost measure, which quantifies the amount of data requests from non-optimal data sources. We have the following main results.

Theorem 7. *At time t, we have*

$$E[R_c(t)] \leq O(ct^z) .$$

Notes on the bound:

- First of all note that $E[R_c(t)] = o(t)$ when $t<1$ and thus $E[R_c(t)]/t\to 0$ as $t\to\infty$. This demonstrates the cost saving property of our algorithm as the average number of redundant data request is converging to 0.
- Clearly z controls the trade-offs between prediction accuracy $r_1(f_{k_1(t)}(t))$ and data acquisition cost regret $E[R_c(t)]$. A higher z leads to a more frequent sampling scheme and thus higher cost regret, while with a small z the sampling is conservative which leads to higher prediction error.

Acknowledgement

This material is based on research sponsored by the NSF under grant CNS-1422211 and the Department of Homeland Security (DHS) Science and Technology Directorate, Homeland Security Advanced Research Projects Agency (HSARPA), Cyber Security Division (DHS

S&T/HSARPA/CSD), BAA 11-02 via contract number HSHQDC-13-C-B0015.

Conclusion

In this paper we consider a problem of finding best set of data for each user to enhance its online learning (be it a classification or prediction problem) performance when facing disparate sources of sequentially arriving samples. We first establish learning error when similarity information among users are known and data can be collected without cost. We then extend the results to the case when such information is unknown a priori. Lastly we propose and analyze a cost-efficient algorithm to help users adaptively distinguish between similar and dis-similar data sources. and aggregate and request data appropriately for the purpose of training predictor and saving budget. We establish its performance guarantee and show the algorithm helps avoid requesting redundant data from sources that are helpless (or even harmful) and thus saves cost.

References

Adams, T. M., and Nobel, A. B. 2010. Uniform convergence of VapnikChervonenkis classes under ergodic sampling. *The Annals of Probability* 38(4):1345–1367.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time Analysis of the Multiarmed Bandit Problem. In *Machine learning*, volume 47, 235–256. Springer.

Crammer, K.; Kearns, M.; and Wortman, J. 2008. Learning from Multiple Sources. *The Journal of Machine Learning Research* 9:1757–1774.

Galesic, M., and Barkoczi, D. 2014. Wisdom of Small Crowds for Diverse Real-World Tasks. In *Available at SSRN 2484234*.

Goldstein, D. G.; McAfee, R. P.; and Suri, S. 2014. The Wisdom of Smaller, Smarter Crowds. In *Proceedings of the fifteenth ACM conference on Economics and computation*, 471–488. ACM.

Hao, S.; Syed, N. A.; Feamster, N.; Gray, A. G.; and Krasser, S. 2009. Detecting Spammers with SNARE: Spatio-temporal Network-level Automatic Reputation Engine. In *Presented as part of the 18th USENIX Security Symposium (USENIX Security 09)*. Montreal, Canada: USENIX.

He, M.; Yang, L.; Zhang, J.; and Vittal, V. A Spatio-temporal Analysis Approach for Short-term Forecast of Wind Farm Generation. *IEEE Trans. Power Syst.*

Hyup Roh, T. 2007. Forecasting the Volatility of Stock Price Index. In *Expert Systems with Applications*, volume 33, 916–922. Elsevier.

Kim, K.-j. 2003. Financial time series forecasting using support vector machines. In *Neurocomputing*, volume 55, 307–319. Elsevier.

Lai, T. L., and Robbins, H. 1985. Asymptotically Efficient Adaptive Allocation Rules. In *Advances in Applied Mathematics*, volume 6, 4–22.

Langford, J., and Zhang, T. 2007. The Epoch-Greedy Algorithm for Multi-armed Bandits with Side Information. In *NIPS*.

Liu, Y., and Liu, M. 2015. Finding Ones Best Crowd: Online Learning By Exploiting Source Similarity. www.umich.edu/~youngliu/pub/aaai16_liu.pdf.

Lu, T.; Pl, D.; and Pal, M. 2010. Contextual Multi-Armed Bandits. In *Journal of Machine Learning Research*, volume 9, 485–492.

Rosenthal, J. S. 1995. Convergence Rates for Markov Chains. *SIAM Review* 37(3):pp. 387–405.

Vapnik, V. N. 1995. *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag New York, Inc.

Example of S

We show $S_{i,j} = 1 - \max_{x\in\mathcal{X},y\in\mathcal{Y}} |Q^i_{x,y} - Q^j_{x,y}|^2$ while setting $\beta_1 := 2\sum_{y\in\mathcal{Y}} y^2$ and $\beta_2 := 2\sum_{x\in\mathcal{X}} \pi^i_x \cdot \sum_{y\in\mathcal{Y}} Q^i_{x,y} \cdot (\sum_{\hat{y}\in\mathcal{Y}} Q^i_{x,\hat{y}}\hat{y} - y)^2$ is a feasible similarity measure. For squared loss the optimal predictor is given by the conditional expectation; we thus have the following:

$$
\begin{aligned}
r_i(f_j) &= \sum_{x\in\mathcal{X}} \pi^i_x \cdot \sum_{y\in\mathcal{Y}} Q^i_{x,y} \cdot (\sum_{\hat{y}\in\mathcal{Y}} Q^j_{x,\hat{y}}\hat{y} - y)^2 \\
&= \sum_{x\in\mathcal{X}} \pi^i_x \cdot \sum_{y\in\mathcal{Y}} Q^i_{x,y} (\sum_{\hat{y}\in\mathcal{Y}} Q^j_{x,\hat{y}}\hat{y} - \sum_{\hat{y}\in\mathcal{Y}} Q^i_{x,\hat{y}}\hat{y} + \sum_{\hat{y}\in\mathcal{Y}} Q^i_{x,\hat{y}}\hat{y} - y)^2 \\
&\le 2\sum_{x\in\mathcal{X}} \pi^i_x \cdot \sum_{y\in\mathcal{Y}} Q^i_{x,y} \cdot (\sum_{\hat{y}\in\mathcal{Y}} Q^j_{x,\hat{y}}\hat{y} - \sum_{\hat{y}\in\mathcal{Y}} Q^i_{x,\hat{y}}\hat{y})^2 \\
&\quad + 2\sum_{x\in\mathcal{X}} \pi^i_x \cdot \sum_{y\in\mathcal{Y}} Q^i_{x,y} \cdot (\sum_{\hat{y}\in\mathcal{Y}} Q^i_{x,\hat{y}}\hat{y} - y)^2 \\
&\le 2\sum_{y\in\mathcal{Y}} y^2 \cdot (1 - S_{i,j}) + \beta_2 \ .
\end{aligned}
$$

Proof of Proposition 3

Suppose $i \in k^*(t)$ and there exists a $n < t$ such that $i \notin k^*(n)$. First consider the following fact: let $0 < \delta < 1$ we have

$$
\begin{aligned}
&|\sqrt{\frac{\log \delta t}{\delta t}} - \sqrt{\frac{\log t}{t}}| \\
&= \frac{1}{\sqrt{\log t} + \sqrt{\frac{\log t + \delta}{\delta}}} \frac{|(1 - 1/\delta)\log t - \delta|}{\sqrt{t}}.
\end{aligned}
$$

Easy to see the first term is strictly decreasing. For the second term since $\sqrt{t}$ is of a higher order compared with $\log t$ we expect this term to be decreasing when t passes certain threshold. Since $i \in k^*(t)$ and $i \notin k^*(n)$ and the fact we proved earlier that the optimal selection is always a continuous group we know $|k^*(n)| < |k^*(t)|$ and denote $\delta := |k^*(n)|/|k^*(t)|$. Therefore reducing $k^*(t)$ to $k^*(n)$ will return a better strategy for time t : compared with time n, the loss from the term $\sqrt{\frac{\log t}{t}}$ to $\sqrt{\frac{\log \delta\cdot t}{\delta\cdot t}}$ is smaller, while the gain in average similarity is the same. Similar arguments hold for the term $(\lambda^i_2)^t$, which is also strictly decreasing with t. Proved.