Sophia Sklaviadis
ssklavi1@jhu.edu

July 17, 2024

Selective Classifiers with Arbitrary Tests

Instance space:

Let X denote the set of possible instances; an instance is the input part x of a training example (x, y) and y is the target label.

Concept class C:

Consider binary functions ("concepts") $f : X \to \{0, 1\}$ that belong to a class C, and are used to label the instances x. The learning algorithm has knowledge of C but not of the specific $f \in C$ that is used to label observations. A class that contains concepts that are too simple may not be expressive enough to describe the data-generating process, while a concept class that is too large would not allow us to design efficient learning algorithms.

Assume that C has VC dimension d.

Data generation:

Let P be a probability distribution over X. The training data is obtained as follows: An instance $x \in X$ is drawn according to P. If $f \in C$ is the target concept-function, the instance x is labeled as $f(x)$ and the learning algorithm observes sample $(x, f(x))$. So, for labeled training data $\{(x_1, y_1), ..., (x_n, y_n)\}$ we assume that $x_1, ..., x_n$ are iid $\sim P$ and are labeled as $(y_1, ..., y_n) = (f(x_1), ..., f(x_n))$.[1]

Our goal is to learn the target function $f \in C$. Consider a $0-1$ loss function $l(y, y') = \mathbb{1}\{y \neq y'\} = |y - y'|$. Consider also a deterministic Emprical Risk Minimization (ERM) oracle that computes $ERM(\mathbf{x}, \mathbf{y}) \in \operatorname{argmin}_{c \in C} \sum_{i=1}^{n} l(y_i, c(x_i))$ for any dataset $\mathbf{x}, \mathbf{y}$. That is, we assume access to a deterministic algorithm that takes as input the labeled training data $\mathbf{x} = (x_1, ..., x_n)$ and labels $\mathbf{y} = (y_1, ..., y_n) = (f(x_1), ..., f(x_n))$ and returns a function h which has 0 training cost

$$h := ERM(\mathbf{x}, \mathbf{y}) = ERM(\mathbf{x}, f(\mathbf{x})) \tag{1}$$

[1] Note the difference with the usual framework where data are generated from a joint distribution D over $X \times Y$ and we factorize $D(x, y) = D(x)D(y|x)$, assuming input space X and target values $Y = \{-1, 1\}$.

and agrees with the unknown f on n points. Essentially there is a "version space" of classifiers with 0 training cost $VS := \{c \in C : c(\mathbf{x}) = \mathbf{y}\}$ and h belongs to it.[2]

Goldwasser et al. 2020 propose a transductive abstention algorithm, i.e. an algorithm that classifies arbitrary test examples in a transductive selective classification setting. Transductive learning refers to the idea of leveraging unlabeled test data during training, i.e. the unlabeled test set is observed alongside the training examples. The test examples are referred to as out of distribution (OOD) test examples, as they come from a distribution Q that is not the same as the distribution of the training examples P.

In summary, the learning algorithm takes as input:

a) training examples from a distribution P over X labeled with some unknown function f that belongs to a class C with finite VC dimension d,

b) arbitrary unlabeled test examples (possibly chosen by an unknown adversary),

and outputs a selective classifier $h|_S(x)$ (or predictor) that abstains from predicting for certain examples.

1 Preliminaries & Background

Definition of the selective classifier:[3] We want to learn an unknown $f \in C$, where C is a family of binary functions of VC dimension d, relative to some distributions P, Q over X. The learner is given examples from P, labeled by $f \in C$, and unlabeled exampled from Q. The learner outputs a selective classifier $h|_S : X \to \{0, 1, \perp\}$. We say that x is rejected if $h|_S(x) = \perp$ and classified when $h|_S(x) = 1/0$. The selective classifier has to learn both a classifier and a subset $S \subseteq X$ of examples to classify.

$$h|_S(x) := \begin{cases} 1 & \text{if } x \in S \text{and } h|_S(x) = f(x) \\ 0 & \text{if } x \in S \text{and } h|_S(x) = 1 - f(x) \\ \perp & \text{if } x \notin S \end{cases} \tag{2}$$

Classification agrees with f in the first case, and it is wrong in the second. An error is a misclassification example that is not rejected.

[2] A. Kalai and Kanade 2021, p.6.

[3] Goldwasser et al. 2020, p.2, `https://arxiv.org/pdf/2007.05145.pdf`; A. T. Kalai and Kanade 2021, p.2f., §1.1.

Note that for the n training data $x_1, ..., x_n$ iid $\sim P$ with labels $y_1, ..., y_n = f(x_1), ..., f(x_n)$ the algorithm first trains a classifier $h := ERM(\mathbf{x}, f(\mathbf{x}))$ agreeing with the unknown f, which has 0 training cost. So, for the training data $x_1, ..., x_n$ we also have $h|_S(x_i) = f(x_i)$ for $i = 1, ..., n$. What we want to do though is to define the selective classifier for all x's that belong to the input space X.

2 Definition of the errors[4]

Let X be the input space, D a distribution defined over X and C the set of functions $X \to \{0, 1\}$ with VC dimension d.

1. For $h, f \in C$, we denote the error of the hypothesis/classifier h wrt the ground truth classifier f by
$$err_D(h) = Pr_{x\sim D}[h(x) \neq f(x)] \quad . \tag{3}$$
For the selective classifier $h|_S : X \to \{0, 1, \perp\}$
$$err_D(h|_S) = Pr_{x\sim D}[h|_S(x) \neq f(x) \wedge h|_S(x) \neq \perp \text{ i.e. } x \in S] \quad . \tag{4}$$
2. We also define the rejection rate of $h|_S$ wrt D
$$rej_D(h|_S) = Pr_{x\sim D}[h|_S(x) = \perp] \quad . \tag{5}$$

3 Goldwasser et al. 2020

Goldwasser et al. 2020 consider two learning settings:

a) "The generalization setting" where the learner is given:

- n iid training examples $\mathbf{x} = (x_1, ..., x_n)$ from a training distribution P labeled by the unknown ground truth classifier $f \in C$ with labels $f(\mathbf{x}) = (f(x_1), ..., f(x_n))$,
- and unlabeled data $\tilde{\mathbf{x}}$ drawn iid from a test distribution $Q \neq P$.
- The algorithm assumes access to an appropriate ERM oracle is that is computed first based on the training data, $h := ERM(\mathbf{x}, f(\mathbf{x}))$.

b) "The transductive setting" where $P = Q$. Here the learner chooses $h := ERM(\mathbf{x}, f(\mathbf{x}))$ using the data from P.

[4] Goldwasser et al. 2020, p.3; A. T. Kalai and Kanade 2021, p.5, §2.

- Then, a white-box adversary that knows h and can see some test data $\mathbf{z}$ from P creates additional arbitrary test data $\tilde{\mathbf{x}}$. So the test data include arbitrary adversarial perturbations.
- The learner looks at the test data at the same time as the training data to come up with a selective classifier.

Goldwasser et al. 2020 provide two algorithms with novel types of guarantees:[5]

- a supervised algorithm called Rejectron that takes as input the labeled training data and the unlabeled test data,
- and an unsupervised algorithm URejectron that uses unlabeled training and test examples.

4 Errors in the generalization (or covariate shift) setting

PQ learning considers two separate rates to measure performance:

i)

$$err_Q = Pr_{\tilde{x}\sim Q}[h|_S(\tilde{x}) \neq f(\tilde{x}) \wedge \tilde{x} \in S] \tag{6}$$

i.e. the misclassification error on future test examples from Q on which the classifier does not abstain.

ii)

$$rej_P = Pr_{x\sim P}[x \notin S] \tag{7}$$

i.e. the fraction of future examples from P on which the classifier abstains.

While apparently intuitive to attempt to use as performance measures err_Q and rej_Q instead of rej_P, the problem is that rej_Q cannot be bound directly. To address this problem Goldwasser et al. 2020 use the following novel idea. The rejection rate wrt P rej_P is used to bound the rejection rate wrt Q as follows:

$$rej_Q \leq rej_P + |P - Q|_{TV} \tag{8}$$

where $|P - Q|_{TV}$ is the total variation distance between P and Q, a measure of non-overlap that ranges from 0, when $P = Q$, to 1, when P and Q have disjoint supports.[6]

[5] Goldwasser et al. 2020, p.2-3; referee 2: https://papers.nips.cc/paper/2020/file/b6c8cf4c587f2ead0c08955ee6e2502b-Review.html

[6] The total variation distance between two distributions P and Q

$$|P - Q|_{TV} = sup_{\text{all events } A}|P(A) - Q(A)|$$

Fact: If X and Y are two random variables with marginal distributions P and Q, then for an event A

$$\begin{aligned}|P(A) - Q(A)| &= |Pr(X \in A) - Pr(Y \in A)| \\ &= |Pr(X \in A, X = Y) + Pr(X \in A, X \neq Y) - Pr(Y \in A, X = Y) - Pr(Y \in A, X \neq Y)| \\ &\leq |Pr(X \in A, X = Y) - Pr(Y \in A, X = Y)| + |Pr(X \in A, X \neq Y) - Pr(Y \in A, X \neq Y)| \\ &= 0 + Pr(X \neq Y)\end{aligned} \tag{9}$$

Lemma 4.1 For any $S \subset X$ and distributions P, Q over X

$$rej_Q(S) \leq rej_P(S) + |P - Q|_{TV} \quad . \tag{10}$$

Proof (?). For $\tilde{x} \sim Q$ and $x \sim P$

$$\begin{aligned}Pr(\text{reject } \tilde{x}) &= Pr(\text{reject } \tilde{x} \text{ and } \tilde{x} = x) + Pr(\text{reject } \tilde{x} \text{ and } \tilde{x} \neq x) \\ &\implies rej_Q \leq rej_P + |P - Q|_{TV}\end{aligned} \tag{11}$$

assuming that $|P - Q|_{TV} = Pr(\tilde{x} \neq x)$ (and ignoring the sup in the definition of total variation distance).

5 Errors in the transductive setting with white box adversary[7]

In the transductive setting there is no Q. The learner first chooses $h := ERM(\mathbf{x}, f(\mathbf{x}))$ with training data $\mathbf{x} \sim P$ and $f(\mathbf{x})$. Then, a true test set $\mathbf{z} \sim P$ is drawn. Based on $\mathbf{x}, \mathbf{z}, f, h$ the adversary modifies any number of examples from $\mathbf{z}$ to create an arbitrary test set $\tilde{\mathbf{x}}$. We consider empirical analogues of err_Q and rej_P

$$err_{\tilde{x}} = \frac{1}{n}|\{i \in [n] : h|_S(\tilde{x}_i) \neq f(\tilde{x}_i) \text{ and } \tilde{x}_i \in S\}| \tag{12}$$

$$rej_z = \frac{1}{n}|\{i \in [n] : z_i \notin S\}| \tag{13}$$

and use them as performance measures.

Again $rej_{\tilde{x}}$ is bounded in terms of rej_z:

$$rej_{\tilde{x}} \leq rej_z + \Delta(z, \tilde{x}) \tag{14}$$

where $\Delta(z, \tilde{x}) = \frac{1}{n}|\{i \in [n] : z_i \neq \tilde{x}_i\}|$ is the transductive analogue of $|P - Q|_{TV}$.

is the largest difference between the probabilities that the two distributions assign to the same event (`https://en.wikipedia.org/wiki/Total_variation`).

[7] Goldwasser et al. 2020, p.8.

6 Guarantees

We consider the algorithm guarantees in terms of the performance measures.

Fundamental Theorem Recall that for any distribution P over examples $x \in X$ and any true classifier $f : X \to \{0, 1\}$ in C of VC dimension d, the so-called Fundamental Theorem of Statistical Learning (FTSL) guarantees that the error rate on future test examples from P (i.e. $P = Q$), of any classifier $h \in C$ that agrees with f on the train data is $\tilde{O}(\frac{d}{n})$:[8]

$$err(h) = Pr_{\tilde{x} \sim P}(h(\tilde{x}) \neq f(\tilde{x})) = \tilde{O}(\frac{d}{n}) \quad .$$

Goldwasser et al. 2020 offer two kinds of guarantees in Theorems 5.2 and 5.4 for the case of the "generalization setting", and in Theorems 5.3 and 5.5 for the case of the "transductive setting".

Guarantee 1 (Thm 5.2):

$$\max\{err_Q, rej_P\} = \tilde{O}\Big(\sqrt{\frac{d}{n}}\Big) \tag{15}$$

Remark: Even when $P = Q$ the guarantees are $\tilde{O}\Big(\sqrt{\frac{d}{n}}\Big)$ compared to $\tilde{O}(\frac{d}{n})$ of the FTSL. Hence, inherent in the algorithm there is a necessary additional cost due to abstaining compared with the common rate $\tilde{O}(\frac{d}{n})$.

Guarantee 2 (Thm 5.4): In the worst case

$$err_Q + rej_P \geq \Omega\Big(\sqrt{\frac{d}{n}}\Big) \quad . \tag{16}$$

7 The Rejectron algorithm

Input:

- train data $\mathbf{x} \in X$ and their labels $\mathbf{y} = f(\mathbf{x})$
- test data $\tilde{\mathbf{x}} \in X$
- error parameter $\epsilon \in [0, 1]$, weight $\Lambda = n + 1$
- classifier $h := ERM(\mathbf{x}, f(\mathbf{x}))$

[8] The soft $\tilde{O}$ hides logarithmic factors, i.e. $f(n) = \tilde{O}(g(n))$ is short for $f(n) = O(g(n) \log^k n)$ for some constant k.

Output: The algorithm defines iteratively a selective classifier $h|_S$, i.e. iteratively chooses a hypothesis h and an acceptance set S.

At iteration $t = 1$:

- start with $S_1 = X$
- choose $c_1 \in C$ to

$$\max_c s_1(c) := err_{\tilde{\mathbf{x}}}(h|_{S_1}, c) - \Lambda \cdot err_{\mathbf{x}}(h, c)$$

  such that c_1 disagrees with $h|_{S_1}$ on $\tilde{\mathbf{x}}$:

$$err_{\tilde{\mathbf{x}}}(h|_{S_1}, c) = \frac{1}{n}|\{i \in [n] : h|_{S_1}(\tilde{x}_i) \neq c(\tilde{x}_i) \text{ and } \tilde{x}_i \in S_1\}|$$

$$h|_{S_1}(\tilde{x}_i) = \begin{cases} h(\tilde{x}_i) & \text{if } \tilde{x}_i \in S_1 \\ \perp & \text{if } \tilde{x}_i \notin S_1 \end{cases},$$

  and c_1 agrees with h on $\mathbf{x}$:

$$err_{\mathbf{x}}(h, c) = \frac{1}{n}|\{i \in [n] : h(x_i) \neq c(x_i) \text{ and } x_i \in S_1\}| \quad .$$

At iteration $t = 2$:

- choose $S_2 = \{x \in X : h(x) = c_1(x)\}$ so that we reject all x's for which c_1 disagrees with h.

8 Lemma 5.1

How do we calculate $c_t = \text{argmax}_c\, s_t(c)$?

Recall that c agrees with h on $\mathbf{x}$, and disagrees with $h|_{S_t}$ on $\tilde{\mathbf{x}}$.

1. Construct an artificial dataset that consists of
   - each training sample, repeated Λ times, and labeled $h(x_i)$, and
   - each test sample $\tilde{x}_i \in S_t$, occurring just once, and labeled $1 - h(\tilde{x}_i)$.
2. Run ERM on this artificial dataset to get a classifier c.
3. The number of errors that the ERM classifier makes on these artificial data is

$$\Lambda \sum_{i \in [n]} |c(x_i) - h(x_i)| + \sum_{i : \tilde{x}_i \in S_t} |c(\tilde{x}_i) - (1 - h(\tilde{x}_i)| \quad = ^9$$

$$\Lambda \sum_{i\in[n]} |c(x_i) - h(x_i)| + \sum_{i:\tilde{x}_i\in S_t} 1 - \sum_{i:\tilde{x}_i\in S_t} |c(\tilde{x}_i) - h(\tilde{x}_i)| \quad = \quad |i \in [n] : \tilde{x}_i \in S_t| - \eta_t(c) \quad .$$

4. The c that minimizes the error in this artificial dataset, maximizes $s_t(c)$.

5. Remark: each S_t is not explicitly stored since $S_1 = X$ could be infinite. Instead we only need to maintain the subset of indices of test examples which are in the acceptance set S_t.

References

[1] Shafi Goldwasser et al. "Beyond perturbations: Learning guarantees with arbitrary adversarial test examples". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 15859–15870.

[2] Adam Kalai and Varun Kanade. "Towards optimally abstaining from prediction with OOD test examples". In: *Advances in Neural Information Processing Systems* 34 (2021).

[3] Adam Tauman Kalai and Varun Kanade. "Efficient Learning with Arbitrary Covariate Shift". In: *Algorithmic Learning Theory*. PMLR. 2021, pp. 850–864.

[9] The 2nd term is the number of times that c disagrees with $1 - h$, and it is equal to the number of times c agrees with h, which is equal to the total $\tilde{x}_i$'s minus the number of times that c disagrees with h. Note that since c and h are binary essentially quantities like $-|c(\tilde{x}_i) - h(\tilde{x}_i)|$ count the number of times there were disagreements.