

Reliable Machine Learning with Refusals

Dennis E. Shasha • 12.XX.2016



Overview

Introduction

Motivation & Problem Setup

Related Work & Conformal P. 

Conjugate Prediction

Offline/Inductive Setup

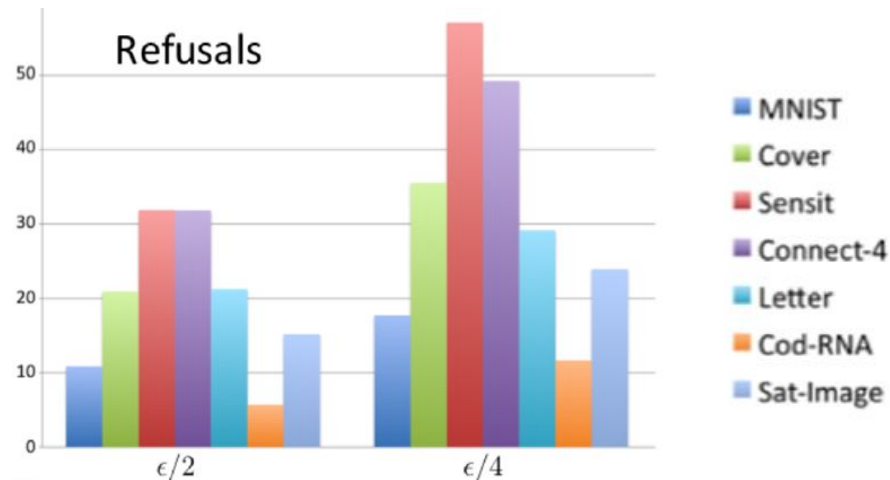
Online/Transductive Setup

Beyond i.i.d.

Combining Specialists with
Error Guarantees

Motivation

- Making a bad decision can be costly, e.g.
 - an unnecessary medical operation
 - a bad trade in finance.



- Our framework takes any machine learning algorithm and reduces its error rate to any target value by allowing refusals.

Problem

Data: $Z_i = (X_i, Y_i) \sim \text{i.i.d. } \mathcal{D}$ for $i = 1, 2, \dots$

Online/Transductive:

- For each t ,
 - Given Z_1, \dots, Z_{t-1}, X_t predict Y_t (OR “*refuse*”)

Offline/Inductive:



- Given Z_1, \dots, Z_m learn a predictor $h: \mathcal{X} \rightarrow \mathcal{Y} \cup \{\text{“refuse”}\}$

Guarantee: $P(\text{Error} \mid \text{Not Refused}) \leq \epsilon$



Existing Work

- Error/Refuse trade-off first studied by Chow (1970)

$$\hat{Y} = \begin{cases} \arg \max_y P(y|X) & \text{if } P(y|X) \geq \tau \\ \text{"refuse"} & \text{otherwise} \end{cases}$$

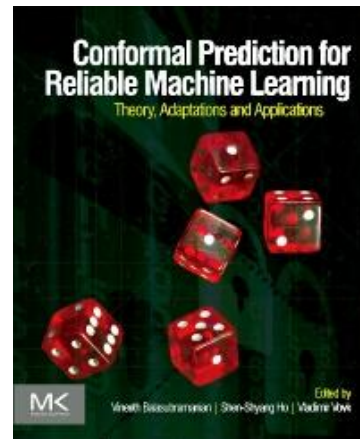
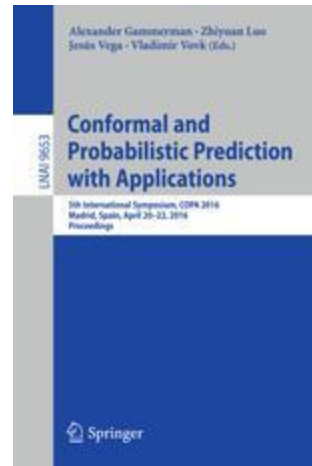
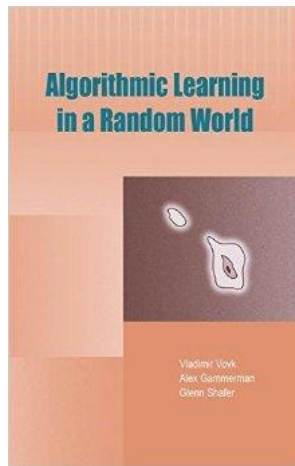
- The optimal  predictor, but requires the knowledge of \mathcal{D}
- Some extensions (Herbei & Wegkamp 2006, 2008):
 - Plug-in rules (see also Denis & Hebiri 2015) 
 - Empirical risk minimizers for $P(\text{Error}) + \alpha P(\text{Refuse})$ (e.g. Cortes, DeSalvo, Mohri 2016)

Existing Work

- Some similar/related problems:
 - Set valued predictors (Sadinle, Lei, Wasserman 2016)
 - Loss functions for information retrieval, e.g. precision/recall (del Coz, Diez, Bahamonde 2009)
 - Confidence based predictors (Lei 2014)
- To our knowledge , all these results are asymptotic results and/or depends on assumptions on the data distribution
 - A notable exception is “conformal prediction” (Vovk, Gammerman, Shafer 2005)
 -  Distribution free tolerance sets (Willks, 1941)

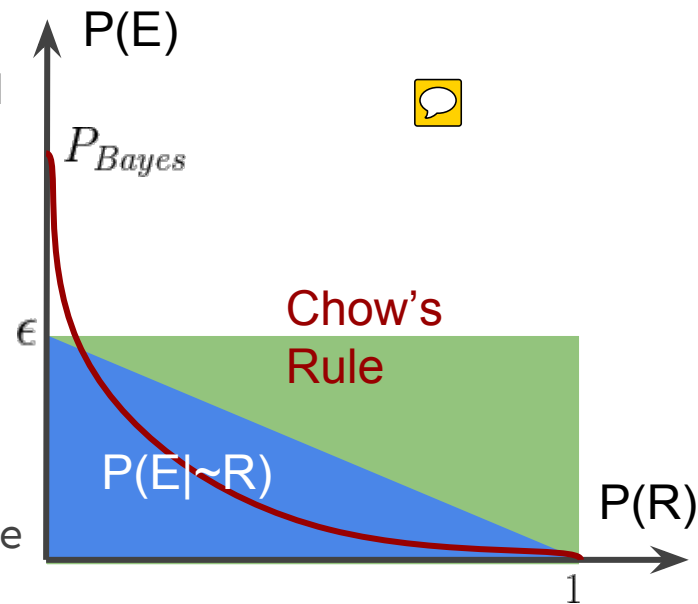
Conformal Prediction

- Introduced by Vovk, Gammerman, and Shafer.
- A meta-algorithm to transform ML algorithms to valid confidence predictors.
- Based on the notion of conformity score.





Our Work

- Based on Conformal Prediction framework
- A meta-algorithm built upon any standard prediction algorithm
 - No assumption on data distribution: exploit the exchangeability.
 - Finite sample guarantee on error probability.
 - Guarantee error rate but may increase refuse rate if data is noisy.

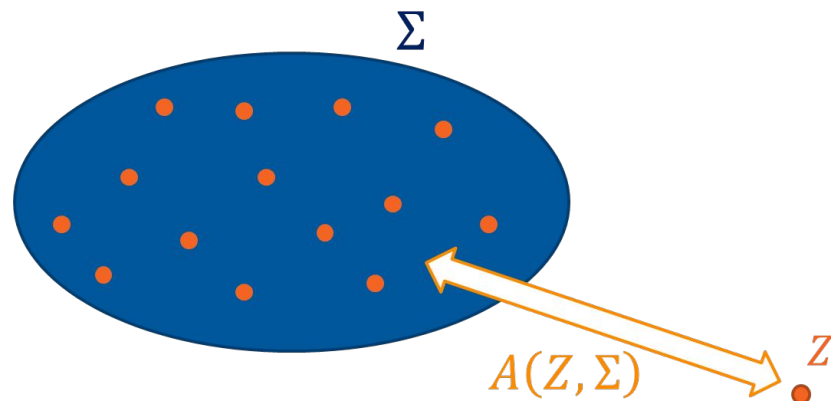


A Note on Efficiency

- Among two predictors with the same error rate, the one  refuses less is more efficient.
- For the rest of the talk, we focus on the probability of error given a sample is not refused; and will not present any formal results on the probability of refusal.
- However, we know from the literature (Vovk, Nouretdinov, Fedorova, Petej, Gammerman 2016), if the base predictor we are building upon is consistent (i.e. it converges to the Bayes' predictor) our algorithm will also  converge to the most efficient predictor (to the Chow's boundary).

Conformity Scores

- A measure of similarity/kinship between a data set (Σ) and a data point (Z)
- Denote with $A(Z, \Sigma)$
- The closer/more similar the point, the larger the score.




Conformity Scores

- How do we measure the conformity score of a point $Z=(X, Y)$ with respect to a set Σ ?
 - Train a classifier on Σ and see how well it performs on Z .
 - **Example:** Train a random forest on Σ and see the fraction of trees that predict the label of X correctly.
- Most machine learning algorithms provides natural conformity measures, e.g.
 - Support vector machines, adaboost, logistic regression, k nearest neighbors, random forests, ...

Offline (Inductive) Conjugate Prediction

Predict or refuse based on the conformity scores as a two step process:

1. **Calibration:** Choose an appropriate acceptance threshold from the domain of conformity scores 
2. **Test:** For given test object X , compute the conformity score of each potential label
 - If more than one label has a score larger than the acceptance threshold, refuse.
 - Otherwise, predict the label with the largest score.

Mechanics of the Algorithm: Example

Assume $Y \in \{\text{low}, \text{medium}, \text{high}\}$.

Train a random forest on a portion of the training set

Predict the label of X using this random forest

Compute the **acceptance threshold** from the rest of the data. 

Suppose it is 0.35, and consider the following scenarios:

- 40% of trees conclude “low”, 30% “med”, 30% “high” \Rightarrow Predict “low”
- 50% of trees conclude “low”, 35% “med”, 15% “high” \Rightarrow Refuse
- 34% of trees conclude “low”, 33% “med”, 33% “high” \Rightarrow Predict “low”

Offline (Inductive) Conjugate Prediction

Calibration step



- Split the training set as **core training** (Σ_{core}) and **calibration sets** (Σ_{cal}).
- Compute scores $\alpha_i(y) = A(\Sigma_{core}, (X_i, y))$ for each X_i in the calibration set, Σ_{cal} .
- Start from the largest **acceptance threshold** (\square) and decrease till we get

$$\frac{\text{Errors on the calibration set} + 1}{\text{Non-refusals on the calibration set} + 1} < \epsilon$$

Note that a larger acceptance threshold implies fewer refusals.

Offline (Inductive) Conjugate Prediction

Test step



- Given the **acceptance threshold** (τ)
 - Compute scores $\alpha(y) = A(\Sigma_{core}, (X, y))$
 - Predict $\hat{Y} = \arg \max_y \alpha(y)$
 - Refuse if $\alpha(y) > \tau$ for more than one y

Error Guarantee



By denoting the predicted test label with $\hat{Y} = h(X)$

Theorem: $P(\hat{Y} \neq Y \mid \hat{Y} \neq ref) \leq \epsilon$

Note, this statement is about the expected performance of the algorithm, not the performance on the particular dataset we have.

However, the inequality holds with high probability as the size of the calibration set becomes larger.



Theorem: $\limsup_{n \rightarrow \infty} 1/n \log P(P(\hat{Y} \neq Y \mid \hat{Y} \neq ref, Z_1^m) > \epsilon) < 0$

where n is the size of the calibration set.

Empirical Results - Datasets

Public datasets from Machine Learning Data Set
Repository (mldata.org)

	<u># of Instances</u>	<u># of Features</u>	<u># of Classes</u>
1. MNIST (scanned handwritten digits),	70000	784	10
2. Cover (dominant forest type based on images),	581012	54	7
3. Sensit (vehicle types from WSN),	98528	100	3
4. Connect-4 (outcome of a multiplayer game),	67557	126	2
5. Letter (letter recognition from pixel displays),	35000	16	26
6. Cod-RNA (coding/non-coding parts of RNA),	488565	8	2
7. Sat-Image (soil type based on satellite images)	10870	36	6

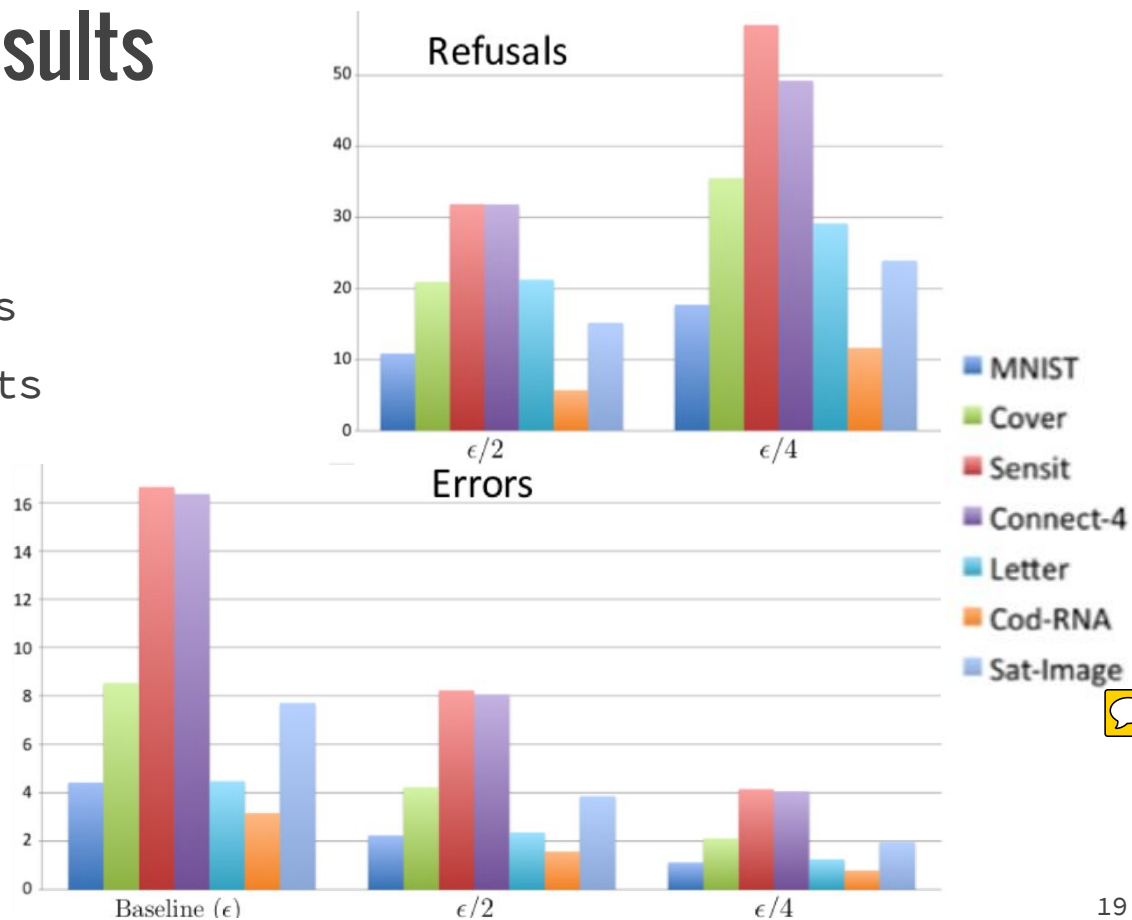
Empirical Results - Experimental Setup

- Base classifier: Random forest (with 100 trees)
- Score: Fraction of trees that predicts the label of X as Y .
- Choose the baseline:
 - Train a random forest over 75% of the data and test on remaining 25%.
- Our meta-algorithm with target rates $\text{baseline}/2$ and $\text{baseline}/4$
 - `core/calibrate/test: 50/25/25`

Empirical Results - Results

Refusal rate increases as the target error rate gets smaller.

- baseline/2: 6-32%,
- baseline/4: 13-57%



Online (Transductive) Conjugate Prediction

- **Intuitive idea:** *split the data stream into two substreams and use each substream to predict the other.*

At each time point with even index $t=2k$:

- Set the core training set $\Sigma_{core} = \{Z_1, Z_3, \dots, Z_{t-1}\}$
- The calibration set $\Sigma_{cal} = \{Z_2, Z_4, \dots, Z_{t-2}\}$

 Then predict \hat{Y}_t following the inductive procedure.

Do the similar for the odd indices.

Error Guarantee

$$E_t = \begin{cases} 0 & \text{if } \hat{Y}_t = Y_t \\ 1 & \text{if } \hat{Y}_t \neq Y_t \end{cases}$$

$$R_t = \begin{cases} 0 & \text{if } \hat{Y}_t \neq \text{"refuse"} \\ 1 & \text{if } \hat{Y}_t = \text{"refuse"} \end{cases}$$

Corollary: For all $t = 1, 2, \dots$ we have $P(E_t = 1 \mid R_t = 0) \leq \epsilon$

- But the errors can be correlated!
 - Suppose for example that each prediction has a probability ϵ of error but if one prediction is in error, then they all are.
 - So, this doesn't say much about observed error rate.

Error Guarantee

However, we know more. The error sequence of each substream is dominated by an independent sequence.

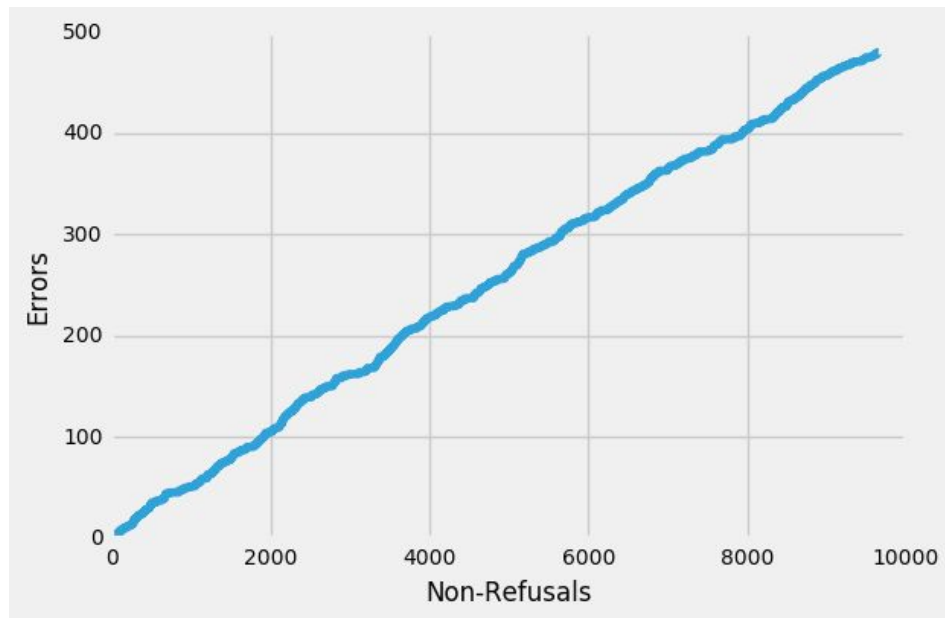
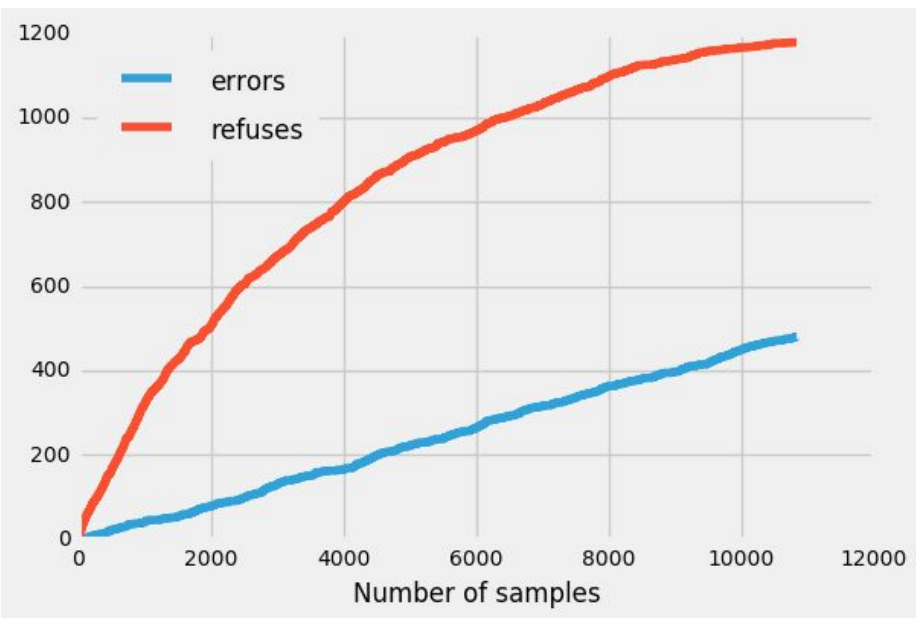
Lemma: For an i.i.d. Bernoulli(\square) sequence B_1, B_2, \dots , there exists a sequence F_1, F_2, \dots such that:

1. $E_t + B_t \cdot R_t \leq F_t$ for all t
2. $F_1, F_3, \dots \sim$ i.i.d. Bernoulli(\square) (same for even indices)

Theorem:
$$\frac{\sum_i E_i}{N - \sum_i R_i} \longrightarrow \epsilon \quad a.s.$$

Empirical Results



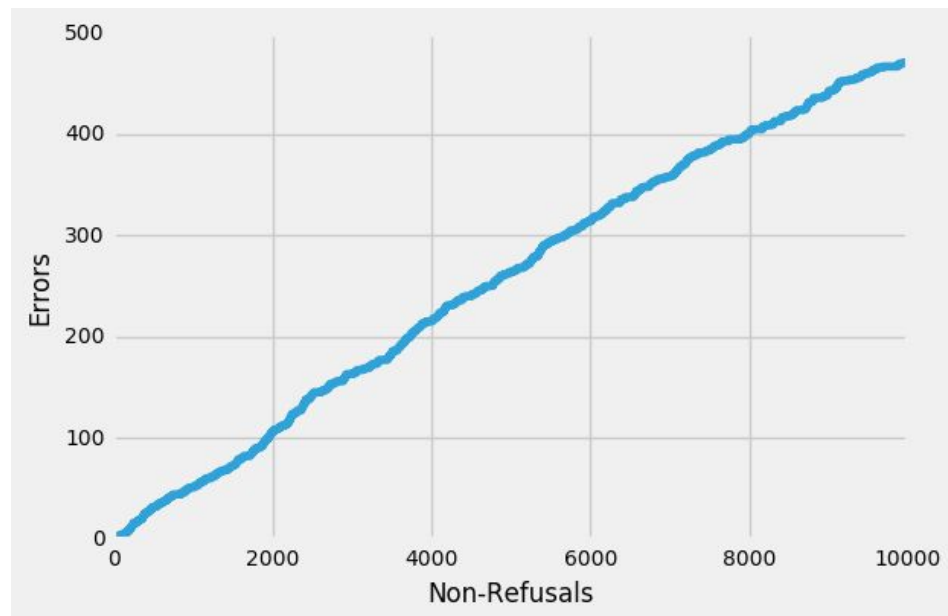
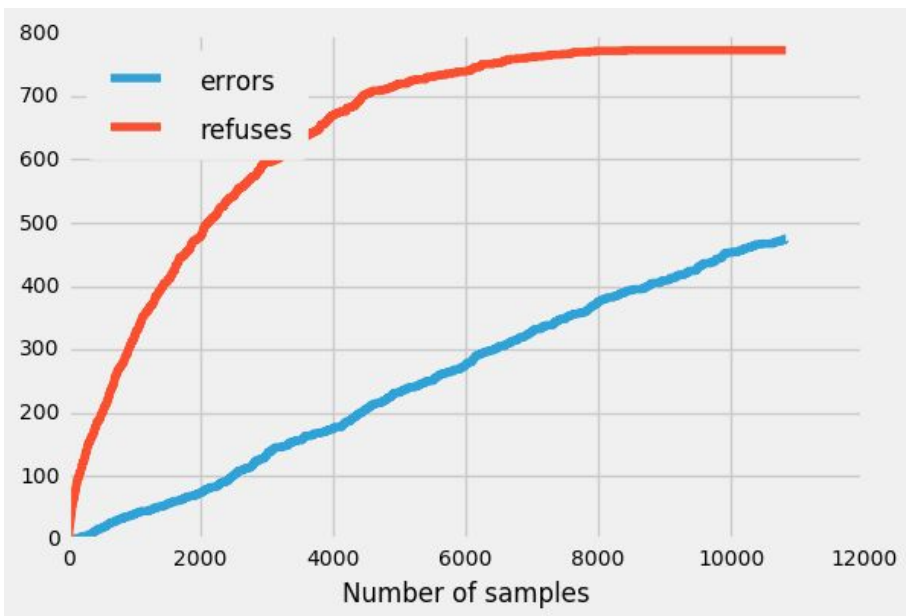


SATI data-set (36 feat/6 labels), RF based scores, Target rate: 0.05

Empirical Results

SATI data-set (36 feat/6 labels), RF based scores, Target rate: 0.02


Empirical Results




SATI data-set (36 feat/6 labels), RF based scores, Target rate: 0.05

Decompose the data into 4 sub-streams instead of 2 (even/odd)

Beyond i.i.d.

- In many applications, assuming the data points are i.i.d. is not reasonable.
 - There might be dependencies among data points
 - Gradual or sudden changes in the data generating process
- One of the most popular approaches when it is  hard to stochastically model the data is prediction with expert advice framework.
- In the following, we adapt the sleeping experts/specialists framework (Freund, Schapire, Singer 1997) to our problem and present asymptotic results by removing the i.i.d. assumption.

Prediction with Expert Advice

- We start with N experts/specialists,
 - For each data point, expert i either predicts the label or refuses to predict.
 - For each false prediction, the expert suffers a unit loss (1)
 - For each expert we maintain a weight w_i to represent its credibility. 
- At each time point t , we choose one of the available predictions made by the experts with probabilities proportional to corresponding experts' weights.

Dummy Expert

As our first step, we add a dummy expert as the $N+1^{\text{st}}$ expert to the ensemble.

- Dummy always makes a dummy prediction and suffers a fixed loss (ϵ) for each step.
- If we choose the dummy at any t , we refuse to make a prediction for Y_t

Updating the Weights

- First initialize all the weights uniformly: $w_i = 1/(N + 1)$
- At each time t , let A_t be the set of active experts.
 1. Preserve the weights of the experts that refused to predict.
 2. Update the weights of the experts that predicted at time t
 - If the i makes an error: $w_i = w_i \cdot e^{-\eta}$
 - Dummy expert: $w_{N+1} = w_{N+1} \cdot e^{-\eta \epsilon}$
 3. Normalize the weights of the active experts ($j \in A_t$):


$$w_j = \frac{w_j}{\sum_{i \in A_t} w_i}$$

Guarantee on Error Rate

Theorem:

The error rate (errors/non-refusals) on the first T data points is less than

$$\epsilon + \sqrt{\frac{(-T \log \sqrt{\delta})^{1/2} + \log(N+1)}{2T^*}}$$

 with probability at least $1-\delta$, where N is the size of the ensemble and T^* is the number of non-refusals.

Remark: “error rate” $-\epsilon = O\left(\frac{T^{1/4}}{T^{*1/2}}\right)$

Guarantee on Refuse Rate

If any of the experts has an error rate less than ϵ , then (asymptotically) we will not reject more often than him.

- $CE(t)$: # of errors on the first t data points
- $CR(t)$: # of refusals on the first t data points
- $CE_i(t)$: # of errors of expert i on the first t data points
- $CR_i(t)$: # of refusals of expert i on the first t data points

Theorem: For any expert j satisfying $\lim_{t \rightarrow \infty} \frac{CE_j(t)}{t - CR_j(t)} < \epsilon$, we have

$$\left(\frac{CR(t) - CR_j(t)}{t} \right)^+ \rightarrow 0 \quad a.s.$$

Conclusion

- If you refuse on occasion, you can reduce errors.
- We give you a systematic meta-algorithm and software for Sci-kit learn that can achieve this.
- If you like this, we can provide to you.