Machine Learning

Yuval Lavie

June 17, 2019

Part I
What is Machine Learning

1. Arthur Samuel - A field of study that gives computers the ability to learn without being explictly programmed.
2. Shai Ben David - A field of study that gives computers the ability to create expertise from experience.
3. Tom Mitchell - A computer program is said to learn from experience E with respect to some task T and some performance measure P if its performance on T as measured by P improves with experience E

Part II
Learning

1 Types of Learning

1. Supervised Learning - A set of answers is available to the learner, and by using these answers he is supposed to create an expertise and answer new questions.

An E-Mail spam program receives a bunch of emails labeled as {Spam/Not Spam} from the user, and uses them to try and label a new received email.

2. Unsupervised Learning - A set of data is available to the learner, and by using this data the learner must create knowledge.

An E-Mail anomaly detection program receives a bunch of emails and tries to label some of them as unusual.

3. Reinforcement Learning - Learning more information on the test examples than exists in the training examples

An E-Mail spam program receives a bunch of emails labeled as spam from the user, and tries to label new emails as spam and also identify malicious senders.

4. Active learning - Interacting with the environment at training time

   An E-Mail spam program actively asking the user to label new emails as {Spam/Not Spam} or even generates new emails itself to try and learn the users preferences

5. Passive learning - Learning only by observing the environment

   An E-Mail spam program can only wait to observe the user's actions on certain emails and use that information to decide.

6. Online Learning - The learner has to respond throughout the learning process.

   A stock broker has to make daily decisions based on the experience he collected.

7. Batch Learning - The learner can output a result only after he had a chance to process a large amount of data

   A data miner will process a huge database before outputing conclusions.

2 Mathematical Frameworks

2.1 The Realizeable Case

We assume that there really exists a deterministic function that defines the labeling for the entire domain space and we wish to find that function or to approximate it.

Input:

1. Let $\mathbb{X} \sim \mathbb{D}$ s.t.
   (a) Domain (Feature) Space : $\mathbb{X}$
   (b) Probability Distribution : $\mathbb{D}$
2. Label Set : $\mathbb{Y}$
3. $\exists f : \mathbb{X} \to \mathbb{Y} | \forall i : f(x_i) = y_i$

1. Training data: $S \sim \mathbb{D}^m =: (X \times Y)^m = \{(x_1, y_1), ...(x_m, y_m)\}$
2. $\mathbb{H}$- Hypothesis class

Output:

1. $h : X \to Y$ - A predictor function that labels each instance $x \in \mathbb{X}$ with a label $y \in \mathbb{Y}$.

Measures:

1. $L_{(\mathbb{D},f)}(h) = \mathbb{P}_{x \in \mathbb{X}}[h(x_i) \neq y_i]$ - a measure of error for the predictor on the real data. cannot be calculated because the learner does not know the distribution or the labeling function.

2. $L_S(h) = \mathbb{P}_{(x,y) \in S}[h(x) \neq y] = \dfrac{|\{(x_i, y_i) \in S : y_i \neq h(x_i)\}|}{|S|}$ - An estimator to the real error.

2.2 The Non-Realizeable Case

We assume that the labels are also generated by a random process, in this scenario two instances of the same values can have a different label!

1. Let $(\mathbb{X}, \mathbb{Y}) \sim \mathbb{D}$ s.t.
   (a) Domain (Feature) Space : $\mathbb{X} \sim \mathbb{D}_x$
   (b) Label Space :$\mathbb{Y} \sim \mathbb{D}_{y|x}$
   (c) Probability Distribution : $\mathbb{D}_{X,Y}$

Input:

1. Training data: $S \sim \mathbb{D}^m =: (X \times Y)^m = \{(x_1, y_1), ...(x_m, y_m)\}$
2. $\mathbb{H}$- Hypothesis class

Output:

1. $h : X \to Y$ - A predictor function that labels each instance $x \in \mathbb{X}$ with a label $y \in \mathbb{Y}$.

Measures:

1. $L_{(\mathbb{D},f)}(h) = \mathbb{P}_{(x,y)\sim\mathbb{D}}[h(x) \neq y]$ - a measure of error for the predictor on the real data. cannot be calculated because the learner does not know the distribution or the labeling function.

2. $L_S(h) = \mathbb{P}_{(x,y) \in S}[h(x) \neq y] = \dfrac{|\{(x_i, y_i) \in S : y_i \neq h(x_i)\}|}{m}$ - An estimator to the real error.

3 Empirical Risk Minimization

We would like to minimize our learners error over the real domain set, but we do not know the distribution of the domain.

1. Realizable Case : $\mathbb{D}, f$ are not known
2. Unrealizeable Case (Agnostic) : $\mathbb{D}_{X,Y}$ is unknown

We therefore try to minimize the empirical error on the training set.

$$ERM_H(S) \in argmin_{h \in H}\Big[L_S(h)\Big]$$

4 Probably Approximately Correct (PAC) Learning

1. δ- The probability to get a misleading sample
2. ϵ- The accuracy of the learner
3. $m_H(\epsilon, \delta)$- function that returns the number of I.I.D samples required to learn a predictor with accuracy ϵ and probability of failure δ

4.1 Realizable PAC Learning

A hypothesis class H is PAC learnable if there exists a function $m_H : (0,1)^2 \to \mathbb{N}$, a learning algorithm A, an unknown distribution $\mathbb{D}$ and a realizable true labeling function $f : X \to Y$ and an I.I.D sample space S such that:
$\forall \epsilon, \delta \in (0,1) \forall \mathbb{D} : |S| > m_H(\epsilon, \delta) \to \mathbb{P}\big[L_{(\mathbb{D},f)}(A(S)) \le \epsilon\big] > 1 - \delta$

4.2 Agnostic PAC Learning

A hypothesis class H is agnostic PAC learnable if there exists a function $m_H : (0,1)^2 \to \mathbb{N}$, a learning algorithm A, an unknown distribution $\mathbb{D}_{(x,y)}$ and an I.I.D Sample Space S such that:
$\forall \epsilon, \delta \in (0,1) \forall \mathbb{D} : |S| > m_H(\epsilon, \delta) \to \mathbb{P}\big[L_{\mathbb{D}}(A(S)) \le min_{h \in H}\big[L_{\mathbb{D}}(h)\big] + \epsilon\big] > 1 - \delta$

4.3 Agnostic PAC Learning for General Loss Function

1. Loss Function - $l : H \times Z \to R_+$
2. Risk - $L_D(h) = \mathbb{E}_{z \sim \mathbb{D}}[l(h, z)]$

A hypothesis class is agnostic PAC learnable with respect to a set Z and a loss function $l : H \times Z \to R_+$, if there exist a function $m_H(\epsilon, \delta) : (0,1)^2 \to \mathbb{N}$, an I.I.D sample space S and a learning algorithm A
such that:
$\forall \epsilon, \delta \in (0,1) \forall Z \sim \mathbb{D} : |S| > m_H(\epsilon, \delta) \to \mathbb{P}\big[L_{\mathbb{D}}(A(S)) \le min_{h \in H}\big[L_{\mathbb{D}}(h)\big] + \epsilon\big] > 1 - \delta$

4.4 Uniform Convergence Learnability

The ERM rule is an agnostic pac learner if a training sample is representative of the data.

1. A training set S is called ϵ-representative with respect to domain Z, hypothesis class H,loss function l, and distribution D if

   $\forall h \in H, |L_S(h) - L_D(h)| < \epsilon$

2. if a sample is $\frac{\epsilon}{2}$-representative then any output of $ERM_H(S)$, namely any $h_S \in argmin_{h \in H}\left[L_S(h)\right]$ satisfies $L_D(h_s) \leq min_{h \in H}\left[L_D(h)\right] + \epsilon$

   $L_D(h_S) \overset{\frac{\epsilon}{2}-Representative}{\leq} L_S(h_S)+\frac{\epsilon}{2} \overset{h_S = \underset{h\in H}{argmin}\left[L_S(h)\right]}{\leq} L_S(h)+\frac{\epsilon}{2} \overset{\frac{\epsilon}{2}-Representative}{\leq} L_D(h) + \epsilon$

3. A hypothesis class H has the Uniform Convergence Property (W.R.T to domain Z and a loss function l) if

   $\forall \epsilon, \delta \in (0,1) \forall Z \sim D \exists m_H^{UC} : (0,1)^2 \rightarrow \mathbb{N} : |S| > m_H^{UC} \rightarrow P\left[|L_S(h) - L_D(h)| \leq \epsilon\right] > 1 - \delta$

5 There Is No Universal Learner (No Free Lunch Theorem)

6 Sample Complexity

1. Every finite hypothesis class is PAC Learnable with sample complexity of $m_H(\epsilon, \delta) \leq \left\lceil \frac{log(|H|/\delta)}{\epsilon} \right\rceil$

   $\delta \leq |H|e^{-\epsilon m} \rightarrow \frac{\delta}{|H|} \leq e^{-\epsilon m} \rightarrow log(\frac{\delta}{|H|}) \leq -\epsilon m \rightarrow -\frac{log(\frac{\delta}{|H|})}{\epsilon} \geq m \rightarrow m > \frac{log(\frac{|H|}{\delta})}{\epsilon}$

2. Every finite hypothesis class enjoys the Uniform Convergence property with sample complexity of $m_H^{UC}(\epsilon, \delta) \leq \left\lceil \frac{log(2|H|/\delta)}{2\epsilon^2} \right\rceil$

7 Glossary

1. ERM - Empirical Risk Minimization : Creating a predictor that minimizes the error on the training sample.

2. $L_S(h)$ - Empirical Error Rate : The rate of error a predictor has on a training sample.

3. $L_{(d,f)}(h)$ - Risk / True Error Rate : The rate of error a predictor has on the distribution and labeling function.

4. Overfitting - A hypothesis fits the training data too well and fails on the real data. $L_S(h) = 0, L_{(d,f)}(h) > \epsilon$

5. Inductive Bias - Choosing a specific Hypothesis Class before seeing the data.

6. Learner's Failure - $L_D(h_s) > \epsilon$

7. Learner's Success - $L_D(h_s) < \epsilon$

8. $m_H(\epsilon, \delta)$ - function that returns the number of I.I.D samples required to learn a predictor with accuracy ϵ and probability of failure δ

9. A sample S is Epsilon-Representative if and only if $|L_S(h) - L_D(h)| < \epsilon$