Progress report - Week 3 - 9/16 - 9/22

Parul Gupta (pargupta@umass.edu)

September 21, 2020

1 Milestones Planned

The following milestone was planned for this week:

1. Implement bound propagation through the parse tree.

2 Milestones Achieved

2.1 Implement bound propagation through the parse tree

2.1.1 STATUS: Done

2.1.2 Things done here

1. Worked through the theory of various inequalities (Hoeffding inequality, t-test) to get the confidence interval for the classes.
2. Worked through how to propagate bounds with various arithmetic operators.
3. Implemented Hoeffding inequality and t-test in Python code.
4. Implemented bound propagation through various operators.
5. Implemented propagation through the expression tree.
6. Implemented and ran tests for both inequalities.

2.1.3 Brief methodology/theory

From previous week, we have expression tree for a given fairness constraint. Now, we want to find the confidence interval[3] of the complete expression (fairness constraint). For this, we need to evaluate bounds for the classes (i.e. TP, FP, TN, FP) and bounds from the operators ('+', '-', '*', '/', 'abs'). We then need to parse it through the parse tree to obtain the resultant confidence interval of the complete expression.

Working with operators:
Consider x = left tree, y = right tree. Let x be bounded by (l_x, u_x) and

y be bounded by ($l\_y,\ u\_y$). Then, following are the bounds across various operators:

1. **Addition**: Cases are -
   - Any of ($l\_x,\ l\_y$) is -infinity: lower bound = -infinity, else: lower bound = $l\_x + l\_y$.
   - Any of ($u\_x,\ u\_y$) is infinity: upper bound = infinity, else: upper bound = $u\_x + u\_y$.
2. **Subtraction**: Cases are -
   - l_x is -infinity or u_y is infinity: lower bound = -infinity, else: lower bound = $l\_x - u\_y$.
   - u_x is infinity or l_y is -infinity: upper bound = infinity, else: upper bound = $u\_x - l\_y$.
3. **Multiplication**: This is more involved with the following cases -
   - Any of x, y are unbounded: confidence interval is unbounded; ie. (-infinity, infinity).
   - x, y are positive:
     If any of ($u\_x,\ u\_y$) = infinity, confidence interval = [$l\_x * l\_y$, infinity). Else, confidence interval = $[l\_x * l\_y, u\_x * u\_y]$.
   - x, y are negative:
     If any of ($l\_x,\ l\_y$) = -infinity, confidence interval = [$u\_x * u\_y$, infinity). Else, confidence interval = $[u\_x * u\_y, l\_x * l\_y]$.
   - One is positive, other is negative (say, x is positive and y is negative):
     If u_x = infinity or l_y = -math.inf, confidence interval = (-infinity, $l\_x * u\_y$]. Else, confidence interval = $[u\_x * l\_y, l\_x * u\_y]$.
     (Similar analysis for the case when x is negative and y is positive)
   - 0 lies in one of the interval bounds (say, 0 lies in x):
     (a) y is positive:
     l_x = -infinity or u_y = infinity, lower bound = -infinity; else, lower bound = $l\_x * u\_y$.
     Any of ($u\_x,\ u\_y$) = infinity, upper bound = infinity; else, upper bound = $u\_x * u\_y$.
     (b) y is negative:
     u_x = infinity or l_y = -infinity, lower bound = -infinity; else, lower bound = $u\_x * l\_y$.
     Any of ($l\_x,\ l\_y$) = -infinity, upper bound = infinity; else, upper bound = $l\_x * l\_y$.

     (Similar analysis can be done for the case when 0 lies in y)
   - 0 lies in both x and y bounds: Any of the lower bound is -infinity or upper bound is infinity, confidence interval is unbounded; ie. (-infinity, infinity). Else, $[min(l\_x * u\_y, u\_x * l\_y),\ max(u\_x * u\_y, l\_x * l\_y)]$.

4. **Division**: Again, this is more involved with numerous cases as follows -

   - x is unbounded or y has 0, confidence interval is unbounded; ie. (-infinity, infinity). This implies that the cases like $infinity/infinity$ or $0/0$ are also considered as unbounded intervals.
   - x, y are positive:
     u_y = infinity, then lower bound = 0; else, lower bound = l_x/u_y.
     u_x = infinity, then upper bound = infinity; else, upper bound = u_x/l_y.
   - x, y are negative:
     l_y = -infinity, then lower bound = 0; else, lower bound = u_x/l_y.
     l_x = -infinity, then upper bound = infinity; else, upper bound = l_x/u_y.
   - x is positive, y is negative:
     u_x = infinity, then lower bound = -infinity; else, lower bound = l_x/u_y.
     l_y = -infinity, then upper bound = 0; else, upper bound = l_x/l_y.
   - x is negative, y is positive: l_x = -infinity, then lower bound = -infinity; else, lower bound = l_x/l_y.
     u_y = infinity, then upper bound = 0; else, upper bound = u_x/u_y.
   - 0 lies in x:
     (a) y is positive:
     l_x = -infinity, then lower bound = -infinity; else, lower bound = l_x/l_y.
     u_x = infinity, then upper bound = infinity; else, upper bound = u_x/l_y.
     (b) y is negative:
     u_x = infinity, then lower bound = -infinity; else, lower bound = u_x/u_y.
     l_x = -infinity, then upper bound = infinity; else, upper bound = l_x/u_y.

5. **Absolute**: This is a unary operator. Cases are -

   - l_x is -infinity or u_x is infinity: confidence interval = [0, infinity).
   - x is positive: confidence interval = $[l\_x, u\_x]$.
   - x is negative: confidence interval = $[-u\_x, -l\_x]$.
   - 0 lies in x: confidence interval = $[min(-l\_x, u\_x), max(-l\_x, u\_x)]$.

Working with inequalities:
We will be implementing 2 different inequalities - Hoeffding's inequality and t-test, to get the confidence intervals.

1. **T-test**[5]:
   Let X_1, X_2, .., X_n be independent and identically distributed random

variables.
Let true mean be μ.
Let sample mean be $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$.
Let sample variance be $\hat{\sigma} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2}$.
Consider δ to be in $(0, 1)$.
If $\bar{X}$ is normally distributed, then by CLT (Central Limit Theorem [2]), we get lower bound as -

$$Pr(\mu \geq \bar{X} - \frac{\hat{\sigma}}{\sqrt{n}} t_{(1-\delta,n-1)}) \geq 1 - \delta$$

Similarly, upper bound can be found as -

$$Pr(\mu \leq \bar{X} + \frac{\hat{\sigma}}{\sqrt{n}} t_{(1-\delta,n-1)}) \geq 1 - \delta$$

For two sided confidence interval, we can say that -

$$Pr(\bar{X} - \frac{\hat{\sigma}}{\sqrt{n}} t_{(1-\delta,n-1)} \leq \mu \leq \bar{X} + \frac{\hat{\sigma}}{\sqrt{n}} t_{(1-\delta,n-1)}) \geq 1 - 2\delta$$

T-test might fail in cases where the sample is concentrated towards one of the lower tail or upper tail of the distribution.

2. **Hoeffding's Inequality**[4]:
Let X_1, X_2, .., X_n be independent and identically distributed random variables.
Let true mean be μ and sample mean be $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$.
Let range of X = [a, b]; ie. $Pr(X \in [a, b]) = 1$.
Consider δ to be in $(0, 1)$.
Then, the lower bound can be found as -

$$Pr(\mu \geq \bar{X} - (b-a)\sqrt{\frac{ln(1/\delta)}{2n}}) \geq 1 - \delta$$

Similarly, upper bound can be found as -

$$Pr(\mu \leq \bar{X} + (b-a)\sqrt{\frac{ln(1/\delta)}{2n}}) \geq 1 - \delta$$

For 2-sided confidence interval, we can use the estimate as -

$$Pr(\bar{X} - 2(b-a)\sqrt{\frac{ln(1/\delta)}{2n}} \leq \mu \leq \bar{X} + 2(b-a)\sqrt{\frac{ln(1/\delta)}{2n}}) \geq 1 - \delta$$

Another way to define this will be -

$$Pr(\bar{X} - (b-a)\sqrt{\frac{ln(1/\delta)}{2n}} \leq \mu \leq \bar{X} + (b-a)\sqrt{\frac{ln(1/\delta)}{2n}}) \geq 1 - 2\delta$$

In the experiment coding, I have used the latter format of the inequality to get 2-sided bound.
Hoeffding's inequality will always hold.

Working through the parse tree:
Union bound or Boole's inequality: For any finite or countable set of events, the probability that at least one of the events happens is no greater than the sum of the probabilities of the individual events [1]. Mathematically,

$$Pr(\cup_{i=1}^{n} A_i) \leq \sum_{i=1}^{n} P(A_i)$$

We will use this inequality to evaluate δ for each node as follows (Assume δ to be significance level for root node):

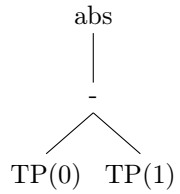
1. **binary operator**: If the node is a binary operator (e.g.: '+'), then it must have 2 children (subtrees). Thus, $\delta_{left} = \delta_{right} = \delta/2$.
2. **unary operator**: If the node is a unary operator (e.g.: 'abs'), then it will have only 1 child (subtree). Thus, $\delta_{child} = \delta$.
3. **leaf node**: As leaf nodes do not have any children, they will directly take the value of δ_{parent}.

2.1.4 Code

The code has been implemented to get confidence interval for operators, confidence interval using hoeffding's inequality as well as t-test and bound propagation through the parse tree.
From the previous week, for the following reverse polish notation (True Positive Rate Sample)-

```
TP(0) TP(1) - abs
```

where $0 = female$ and $1 = male$; the parse expression tree was:



We passed data in the form of true value, Y, predicted value, $\hat{Y}$ and sensitive attribute, T. These were *numpy.Series* (1-D array) to get the estimate and the respective confidence interval. Y and $predicted_Y$ are assumed to be 0,1 binary classification.

Test run:

```
Inputs:
Y = [0, 0, 0, 1, 1, 1]
pred_Y = [1, 1, 1, 1, 1, 1]
T = [0, 1, 0, 1, 0, 1]
expression = TP(0) TP(1) - abs
```

Confidence interval of TP(0) and TP(1) for the above input as:

```
Confidence Interval for TP(0) with delta = 0.05:
 T test: [-1.0431610698908815, 1.709827736557548]
 Hoeffding Inequality: [-0.16595537892566514, 0.8326220455923318]
Confidence Interval for TP(1) with delta = 0.05:
 T test: [-0.021580534945440766, 1.354913868278774]
 Hoeffding Inequality: [0.16737795440766817, 1.1659553789256651]
```

Confidence interval for the above fairness constraint and input values:

```
Confidence Interval for expression tree with delta = 0.05:
 T test: [2.7091015890777403, 3.375768255744407]
 Hoeffding Inequality: [0.896293151371312, 1.562959818037979]
```

Expression tree is parsed $\delta = 0.05$. δ doesn't change for child of 'abs' operator. As '-' is a binary operator, TP(0) and TP(1) will get $\delta = 0.025$.
For **T-test**, confidence interval for TP(0)= [-1.695, 2.362], TP(1) = [-0.347, 1.681]. On '-', confidence interval = [-3.376, 2.709]. Finally, on taking 'abs', confidence interval = [2.709, 3.376].
For **Hoeffding's inequality**, confidence interval for TP(0)= [-0.281, 0.948], TP(1) = [0.052, 1.281]. On '-', confidence interval = [-1.563, 0.896]. Finally, on taking 'abs', confidence interval = [0.896, 1.563].

Code is present in a private github repo: fair-work

Next steps: Add more unit test cases for the code. Also, test the whole workflow on bigger synthetic dataset and with more fairness constraints.

References

[1] Boole's inequality, https://en.wikipedia.org/wiki/boole%27s_inequality.

[2] Central limit theorem, https://en.wikipedia.org/wiki/central_limit_theorem.

[3] Confidence interval, https://en.wikipedia.org/wiki/confidence_interval.

[4] Hoeffding's inequality, https://en.wikipedia.org/wiki/hoeffding%27s_inequality.

[5] T-test, https://en.wikipedia.org/wiki/student's_t-test.