CS *521* Project

Achieving Fine Grained Control Over Incidence of False-positives and False-negatives in NLP Classification Tasks using Methods Inspired by Game Theory

Akshay Channesh
achann4 at uic dot edu

Introduction

- Original Plan: *Use Game Theory for work in Semantics.*
- Hard, and too complex. Too much time spent and nothing to show for it.
- A lower hanging fruit was found!
- Still interesting and still uses

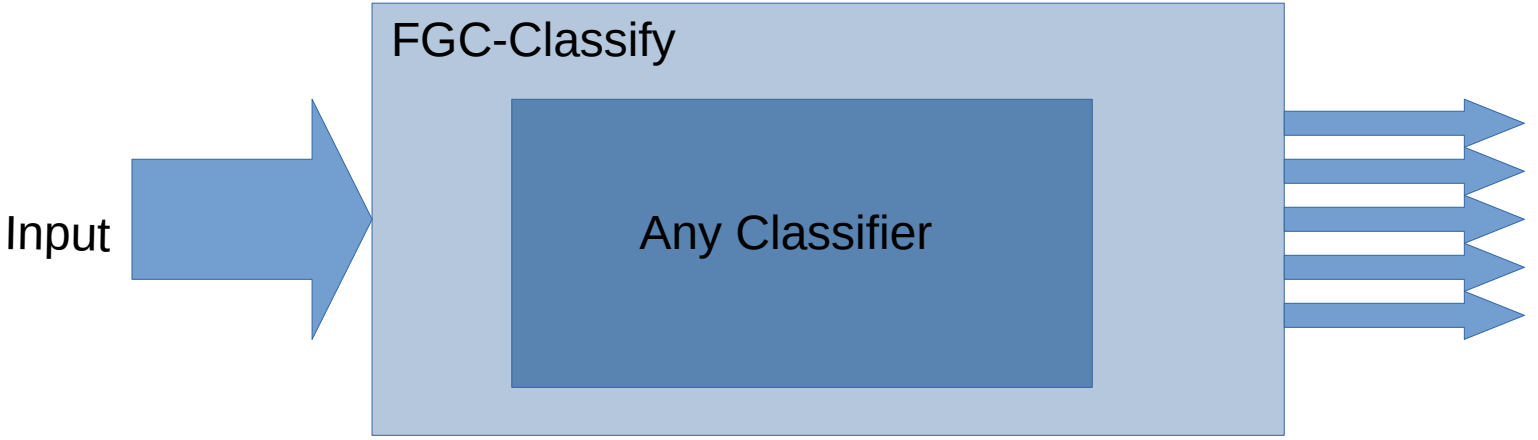
  Game Theory and Equilibrium methods

The Problem

- Classifiers are somewhat rigid.
- What if I want to reduce/increase the number of False Positives or False Negatives, by trading off on other parameters like accuracy?
- In some cases this is very important!
  - Eg. “Alexa! Don’t you listen to me unless I call you!” [False Positives are dangerous]
  - Spam: I really don’t want my College Acceptance Letter to end up in spam.
  - More SMS spam/ Less SMS spam. No one close to you uses SMS so might as well block all.

Methods

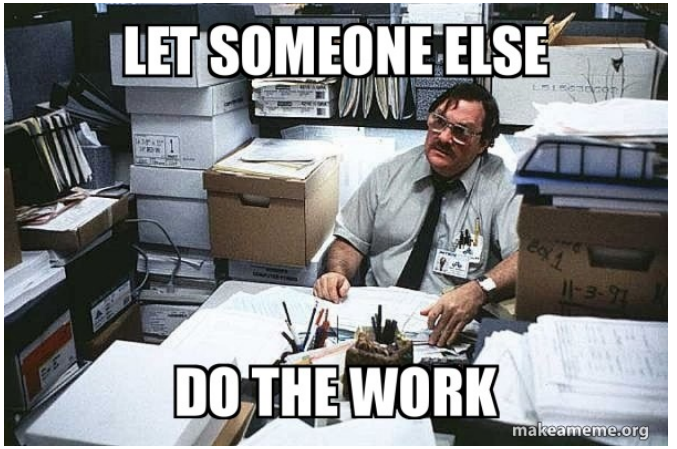
- Algorithm FGC-Classify: A meta algorithm that runs a game over any existing classifier and gives you control



Closest existing work

- Scikit Learn “`class_weight`” parameter.
- Not classifier agnostic.

  Must be handled by the classifier itself.
- Very restricted context.



Inspiration

- Communication and Content. P. Parikh. 2019
- p.2*14* The Author remarks in passing. However does not get into the how.
- … Clearly, the Nash solution involves comparing expected values of payoffs rather than pure probabilities. By adjusting the payoffs we can get different, more fine-grained results. For example, ordinarily we would want false positives (i.e. classifying some text as spam when it is not spam) to be penalized more heavily. If it is assumed that the sender in s is *spamming* and in s' is *not spamming* sufficiently low, that is, then such a heavier penalty can be realized by making … the cost of interpreting a text that is not spam as spam high.

The Algorithm: Overview

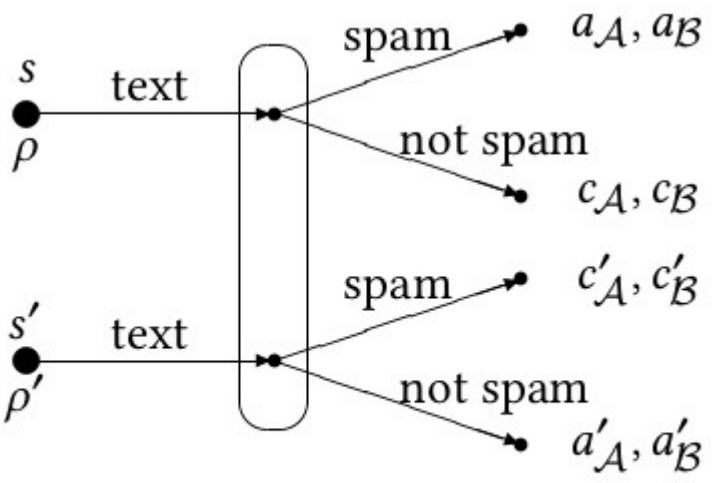


Figure 12.2: Text classification game g_{tc}

- FGC-Classify(Input, Utils)
  - Run Classify(Input)
  - Select Best_Response based on Payoffs
  - Return Best_Response

Utilities as Hyper-parameters

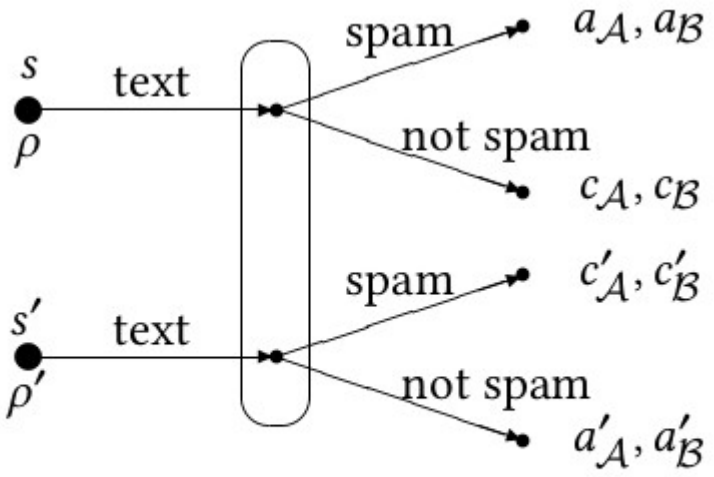


Figure 12.2: Text classification game g_{tc}

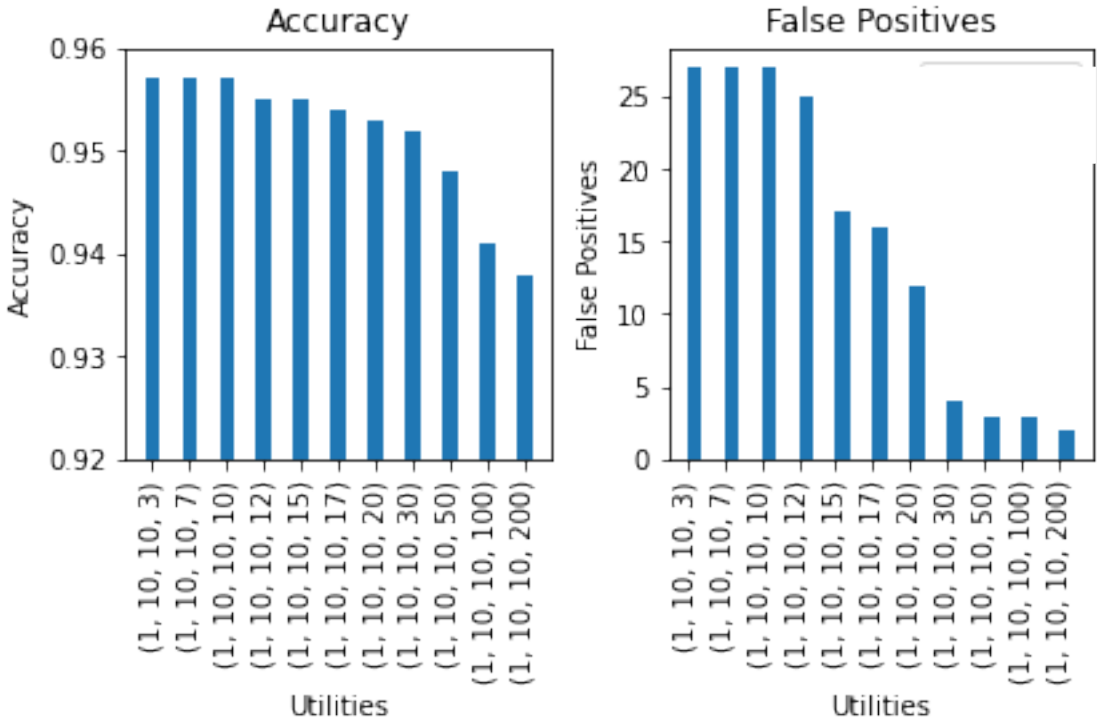
Evaluation

- Naive Bayes Classifier. Follows the tutorial by KDnuggets https://www.kdnuggets.com/2020/07/spam-filter-python-naive-bayes-scratch.html
- SMS Spam
- Dataset: SMS Spam Collection v.1 dataset. The corpus has been collected by Tiago Agostinho de Almeida (http://www.dt.fee.unicamp.br/~tiago) and José María Gómez Hidalgo (http://www.esp.uem.es/jmgomez), and can be found here: (https://archive.ics.uci.edu/ml/datasets/sms+spam+collection#)

```
Correct: 5170
Incorrect: 235
Accuracy: 0.9565217391304348
False positives: (when a true ham is falsly flagged as spam) 27
False negatives: (when a true Spam is falsly allowed as ham) 207
```

Reducing False Positives

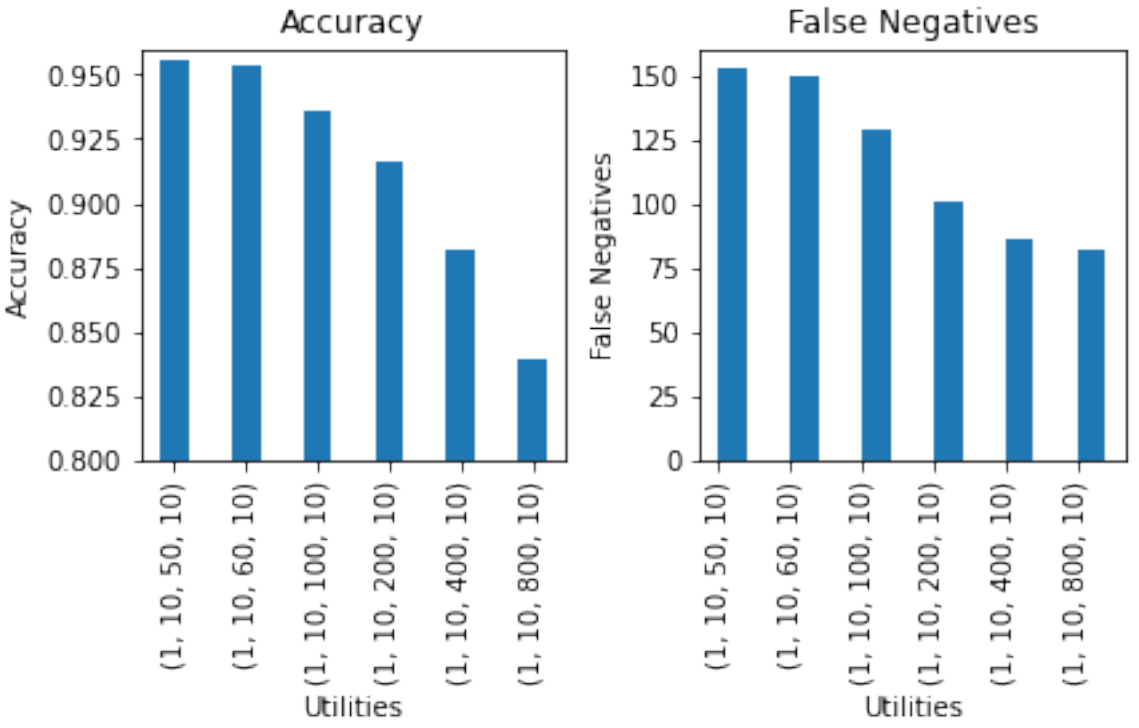
```
Correct, Incorrect, Accuracy, F_Pos, F_Neg
[(5170, 235, 0.9565217391304348, 27, 207),
 (5170, 235, 0.9565217391304348, 27, 207),
 (5170, 235, 0.9565217391304348, 27, 207),
 (5162, 243, 0.9550416281221091, 25, 217),
 (5161, 244, 0.9548566142460685, 17, 226),
 (5159, 246, 0.9544865864939871, 16, 229),
 (5153, 252, 0.9533765032377428, 12, 239),
 (5143, 262, 0.9515263644773359, 4, 257),
 (5126, 279, 0.9483811285846439, 3, 275),
 (5087, 318, 0.9411655874190564, 3, 314),
 (5068, 337, 0.937650323774283, 2, 334)]
```



F_Pos reduces
Accuracy drops slightly
F_Neg increases

Reducing False Negatives

```
Correct, Incorrect, Accuracy, F_Pos, F_Neg
[(5167, 238, 0.9559666975023127, 84, 153),
 (5157, 248, 0.9541165587419056, 97, 150),
 (5058, 347, 0.9358001850138761, 217, 129),
 (4950, 455, 0.9158186864014801, 353, 101),
 (4769, 636, 0.8823311748381129, 549, 86),
 (4536, 869, 0.8392229417206291, 786, 82)]
```



F_Neg reduces
Accuracy drops significantly
F_Pos shoots up dramatically

Discussion of Results

- Utilities are Hyper-parameters
- Good part is that Hyper-parameter tuning is very easy.
- Utilities are linear. And robust to

  Affine Transformations of the form $aX + B$
- *Utility Theory:* Very well understood

  https://en.wikipedia.org/wiki/Von_Neumann%E2%80%93Morgenstern_utility_theorem
- Reducing incidence of any one kind of classification

  is as easy as increasing the weight of other.

Conclusion & Future Work

- Wraps around any existing classifier. [Black Box]
- Selects output using Best_Response based on Utilities.
- Future Work: Word Sense Disambiguation is a classification task as well.
- Extends easily to Multinomial Classification