Evaluation

CS 685, Spring 2020

Advanced Topics in Natural Language Processing

http://brenocon.com/cs685

https://people.cs.umass.edu/~brenocon/cs685_s20/

Brendan O'Connor

College of Information and Computer Sciences

University of Massachusetts Amherst

Announcements

- HW2
- Today
  - 1. Agreement rates
  - 2. Evaluation metrics
  - 3. Statistical testing

Annotations quality

- Measurement theory from social sciences asks about
  - **Validity**: is it right?
    - Do the annotations correspond to the deeper concept you care about? ("Construct validity") For your application, analysis goal, etc.
  - **Reliability**: is it repeatable?

[Quinn et al. 2010]

Reliability

- The annotations you got - **are they repeatable**?

  Interannotator Agreement Rate

  - How much do two humans agree on labels?
    - Simple quantitative metric! Next slide.
  - Difficulty of task. Human training? Human motivation/effort?
- Goal: get the human performance "upper bound"
  - Does human agreement rate represent an upper bound for machine performance?

NLP System

<u>Can</u> Beat human perf

- Humans make mistakes
- Highly objective task
- Non-annotated labels ... vs ... human-given labels
- Expert knowledge?

<u>Can not</u> beat hum. perf

- Supervision with low-agr rate labels

Measuring agreement rates

- Assume two annotators both judge a set of items
- **Agreement rate**: proportion of time two annotators agree
  - i.e., accuracy of one annotator matching the other
- Chance-adjusted agreement
  - If some classes predominate, raw agreement rate may be misleading
  - Many similar measures for this: Cohen's kappa, Krippendorff's alpha, etc.
- Cohen's kappa

Inter-Anno

IAA

ICR

chance (random) agr rate

$= \sum_k P(k)P(k) = \sum_k (P(k))^2$

Obs. agr rate

$$\kappa = \frac{\text{agreement} - E[\text{agreement}]}{1 - E[\text{agreement}]}$$

$\Rightarrow .95^2 + .05^2$

if agr = E[agr] $\Rightarrow$ Cohen $\kappa = 0$

- [GO TO SPREADSHEET]

Annotation process

- To pilot a new task, requires an iterative process
  - Look at data to see what's possible
  - Conceptualize the task, try it yourself
  - Write annotation guidelines
  - Have annotators try to do it. Where do they disagree? What feedback do they have?
  - Revise guidelines and repeat
- If you don't do this, your labeled data will have lots of unclear, arbitrary, and implicit decisions inside of it
- **Annotated data is at the heart of real-world NLP applications!!!!**

Hard Classification Metrics

- Many different metrics can be calculated from the **confusion matrix**

$$\text{Acc} = \frac{1}{N}\sum_i 1\{\hat{y}_i = y_i\} = \frac{TP+TN}{N} = 1 - \frac{FP+FN}{N}$$

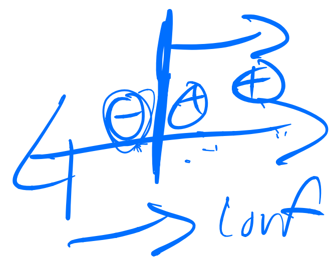
gold \ pred	1	0
1	True Pos	**False Neg**
0	**False Pos**	True Neg

FN

FP

$$\text{Prec} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$



- Many different metrics can be calculated from the **confusion matrix**

http://brenocon.com/confusion_matrix_diagrams.pdf

Trading off FPs vs. FNs

- All ML-based classifiers use a **confidence threshold**
  - Trades off between false positive vs. negatives
- In NLP, Precision and Recall are usually used - ignore TNs (makes sense for a rare class)
- Which matters more? Application-specific!
- Arbitrary, but common, answer: **F1 score**
  - Harmonic mean and set overlap interpretations

$\arg\max_k P(y=k|x)$

Hard thresh

$$\hat{y} = 1\{P(y=1|x) > 0.5\}$$

Vary thresh

$$= 1\{P(y=1|x) > t\}$$

$t=0.9$ high conf!

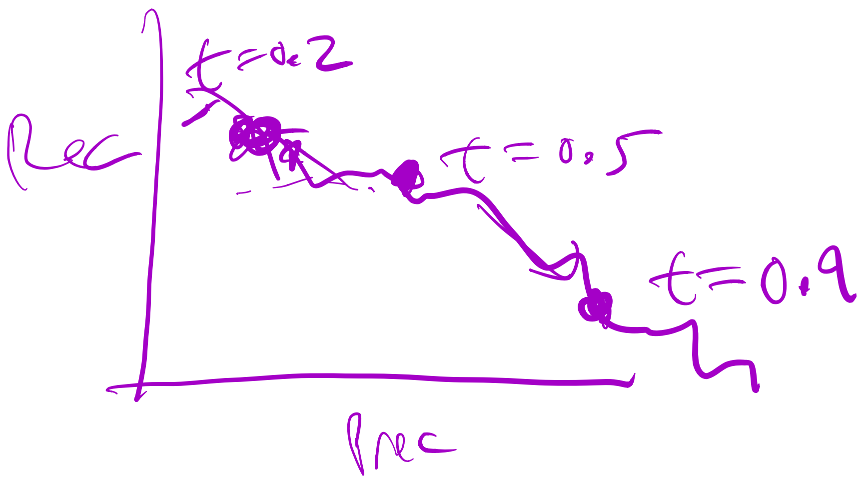
Prec ↑

Rec ↓

https://brenocon.com/blog/2012/04/f-scores-dice-and-jaccard-set-similarity/

Other evaluation metrics

- Probabilistic predictions: you can evaluate log-likelihood on the test set, too!
- Multiclass: do you care about rare classes as much as common classes?
    - Care about examples equally: “micro-averaged” prec/rec/f1
      directly use overall TN/FP/FN counts
    - Care about classes equally: “macro-averaged” prec/rec/f1
      unweighted mean of per-class metrics
- **Precision-Recall Curve**: each decision threshold defines a particular precision/recall tradeoff. Area Under PR Curve is one of several threshold-free metrics (ranking metrics)



Statistical variability in NLP

- How to trust experiment results, given many ***sources of variability***?
- If you _____, would you get the same result?
  - If you sampled the text data again
    - randomness in data sampling
  - If you collected annotations again
    - randomness in human behavior
  - If you ran your algorithm again
    - randomness in computational model (neither NB or logreg have this…)

Statistical Testing

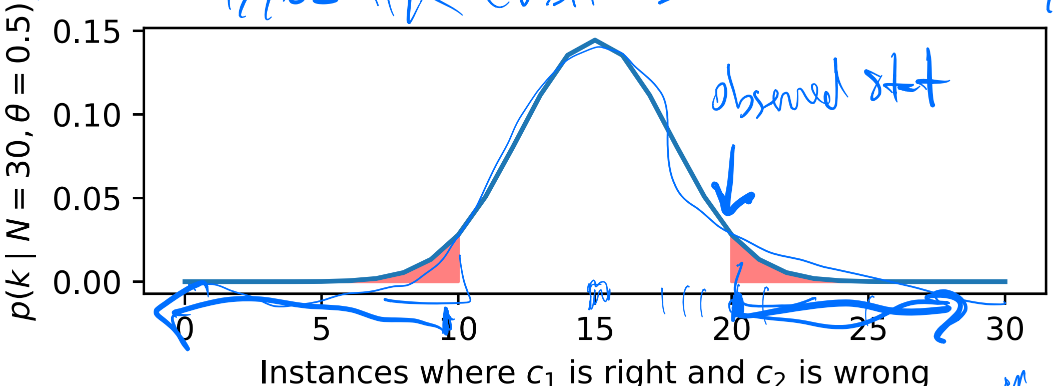
- A way to formalize analysis of variability
- Vast majority of work only looks at resampling textual examples
- Two types of analyses
    - p-values for Null Hypothesis Testing
        - one method: binomial test
    - confidence intervals
        - one method: bootstrapping

Null hypothesis test

pvalue = red areas below

- Must define a <u>null hypothesis</u> you wish to ~disprove
- pvalue = Probability of a result as least as extreme, if the null hypothesis was active
- Example: paired testing of classifiers with exact binomial test (R: *binom.test*)

Null Hypo: the classifiers have same perf



6.8% whole

3.3% or

long?

$$p_{\text{Binom}}(k; N, \theta) = \binom{N}{k}\theta^k(1-\theta)^{N-k}$$

The Bootstrap

- One of many tests - very flexible and conceptually sound (but use others as appropriate!)
- Idea: We want to know how much different another sampled dataset could have been. Simulate this by drawing a new dataset, *with replacement*, from your current one!
  - The *distribution* of bootstrapped evaluation scores is of interest and provides e.g. a 95% confidence interval: its [2.5%ile, 97.5%ile]
  - You can use *any* evaluation method you want!
    - Weird things like F1 score, AUPRC, etc.
    - The *difference* in scores between two different classifiers on the same data

- [GO TO SPREADSHEET]

Algorithm 7 Bootstrap sampling for classifier evaluation. The original test set is $\{\boldsymbol{x}^{(1:N)}, \boldsymbol{y}^{(1:N)}\}$, the metric is $\delta(\cdot)$, and the number of samples is M.

---

procedure BOOTSTRAP-SAMPLE($\boldsymbol{x}^{(1:N)}, \boldsymbol{y}^{(1:N)}, \delta(\cdot), M$)
    for $t \in \{1, 2, \ldots, M\}$ **do**
        for $i \in \{1, 2, \ldots, N\}$ **do**
            $j \sim \text{UniformInteger}(1, N)$
            $\tilde{\boldsymbol{x}}^{(i)} \leftarrow \boldsymbol{x}^{(j)}$
            $\tilde{\boldsymbol{y}}^{(i)} \leftarrow \boldsymbol{y}^{(j)}$
        $d^{(t)} \leftarrow \delta(\tilde{\boldsymbol{x}}^{(1:N)}, \tilde{\boldsymbol{y}}^{(1:N)})$
    return $\{d^{(t)}\}_{t=1}^{M}$
