Classification Evaluation

CS 485, Fall 2024
Applications of Natural Language Processing

Brendan O'Connor
College of Information and Computer Sciences
University of Massachusetts Amherst

Office hours & your TA!

Hui Wei

PhD student in Computer Science

College of Information & Computer Sciences

University of Massachusetts, Amherst

Email: huiwei@cs.umass.edu

[About Me] [Research Interests] [Experiences] [Honors and Awards] [Services] [Miscellaneous] [Hobbies]

- **Brendan: Tuesdays 1-2pm, CS 238**. That is, starting in my office ~15 minutes after class ends. For quick questions, feel free to ask right after lecture.)
- **Hui: Wednesdays 11am-12pm**, LGRT T220.

- See Piazza pinned post for latest information & zoom link

Evaluation

- Evaluation
  - Test on held-out data
  - What precise metrics can we use? What makes sense for
    - Unbalanced data
    - Multiclass
  - How can we trade off types of errors at runtime, after a model is trained?
- Annotation

False Pos vs False Neg

- Definitions

y : gold-std. label

$\hat{y}$: predicted label

$Acc = \frac{1}{N} \sum_i 1\{y_i = \hat{y}_i\}$

Spam classif ?

False Pos : $\hat{y} = 1$ but $y = 0$

False Neg : $\hat{y} = 0$ but $y = 1$

- Are the tradeoffs the same for different applications or tasks?

Asymm. Costs

Health Tests

Credit Card Fraud

Evaluation metrics

$$P(y=1 \mid \hat{y}=1) = \frac{tp}{tp+fp} = 1 - \frac{fp}{tp+fp}$$

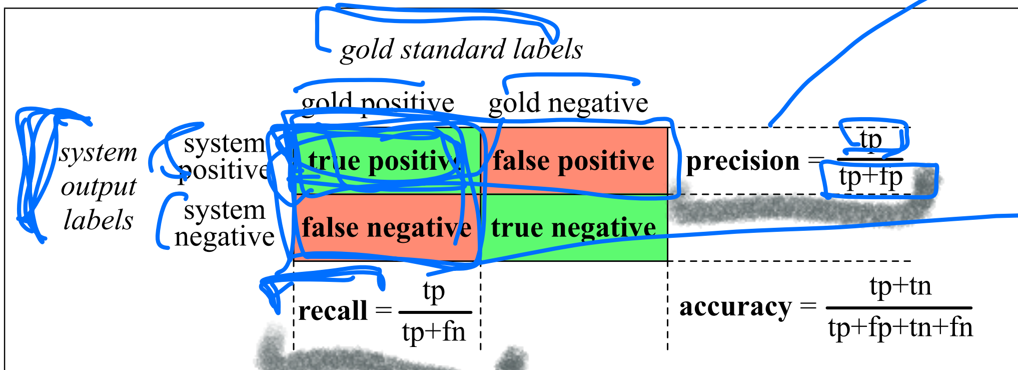


Figure 4.4 A confusion matrix for visualizing how well a binary classification system performs against gold standard labels.

$$P(\hat{y}=1 \mid y=1) = \frac{tp}{tp+fn} = 1 - \frac{fn}{tp+fn}$$

- Accuracy:
    - But do we care about false positives and negatives equally?
    - What about rare classes?
- Precision, Recall, F1

Ignore TNegs!

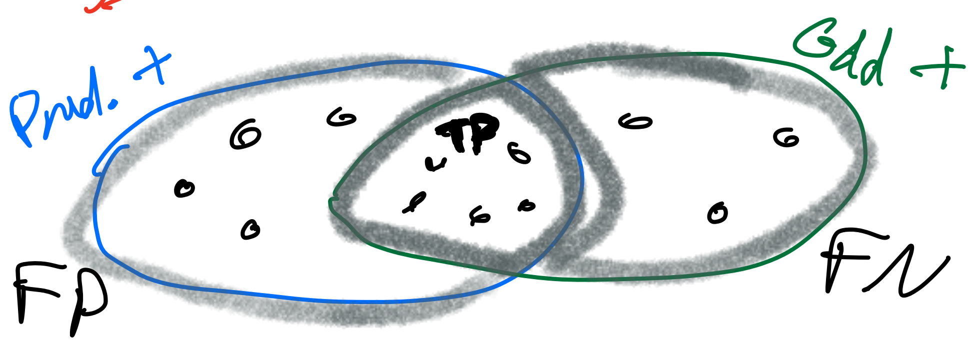
TN common for rare class pred.

Precision, recall, F1

F_1 - Score: Harmonic mean of prec & rec

$$F_1 = \frac{1}{\frac{1}{2}\left[\frac{1}{P} + \frac{1}{R}\right]} = \frac{2 \cdot P \cdot R}{P + R}$$

Weighted F_β-Score: care more abt P or R



Decision threshold

- Problem: you'd like a higher precision model (for class SPAM), and willing to sacrifice recall.
- Solution: predict SPAM more conservatively: only if probability exceeds a threshold

Default decision rule: $\hat{y} = \mathbb{1}\{ p(y=1|x) > 0.5 \}$

$$\hat{y} = \mathbb{1}\{ p(y=1|x) > t \}$$

Thresholded decision rule:

$t = 0.9$

Visualizing a classifier in feature space

"Bias term"

Feature vector $x = (1,\ \text{count "happy"},\ \text{count "hello"})$

Weights/parameters $\beta = (const,\ large,\ small)$

50% prob where

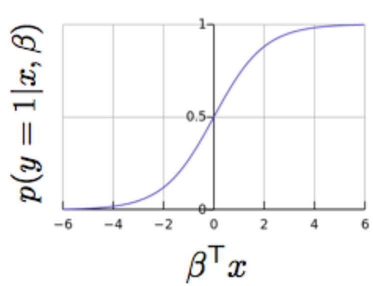
$$\beta^\mathsf{T} x = 0$$

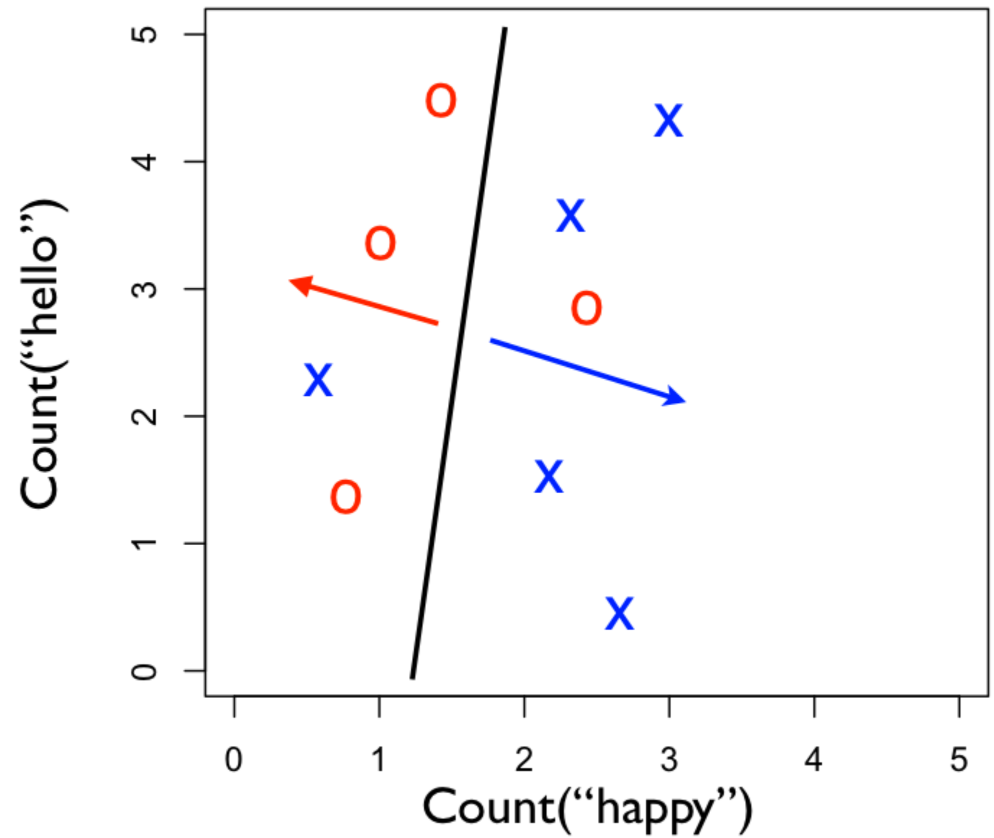
Predict y=1 when

$$\beta^\mathsf{T} x > 0$$

Predict y=0 when

$$\beta^\mathsf{T} x \leq 0$$





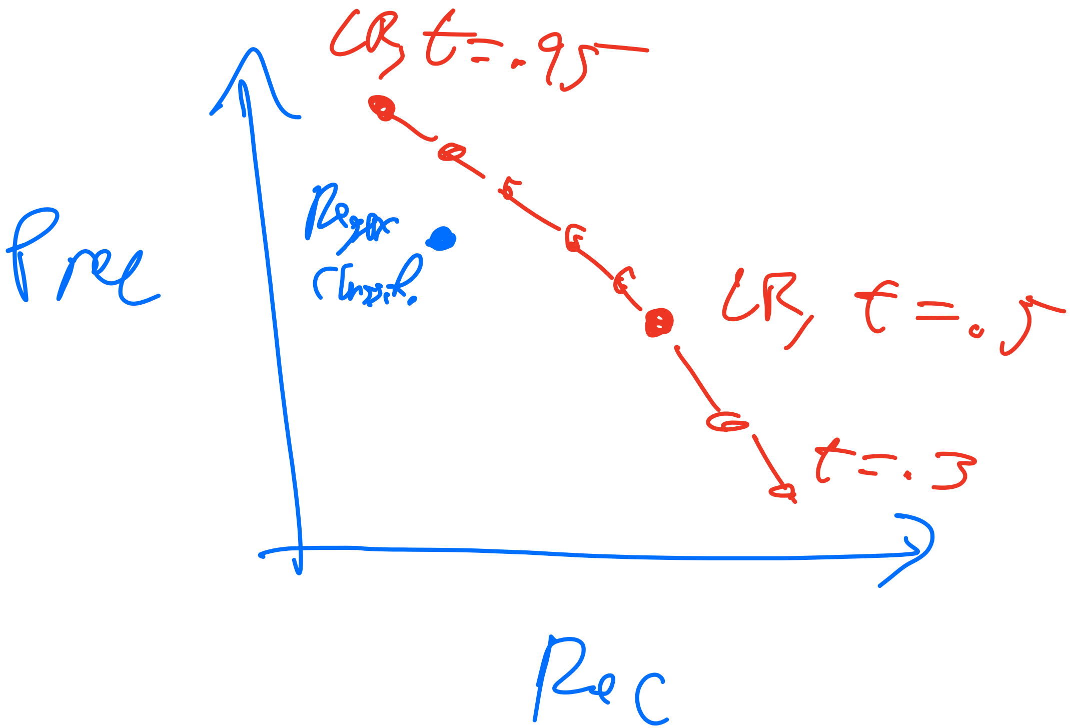
Decision threshold

- How do errors change as threshold *increases*?

Precision-Recall curve

- Different models may trade off precision and recall
- For a single model, different decision thresholds may trade off precision and recall
- View them jointly with a **precision-recall curve**

Precision-Recall curve



Multiclass metrics

- Every class has its own TP, FP, FN counts!

```
from sklearn.metrics import classification_report
>>> y_true = [0, 1, 2, 2, 0]
>>> y_pred = [0, 0, 2, 1, 0]
>>> target_names = ['class 0', 'class 1', 'class 2']
>>> print(classification_report(y_true, y_pred, target_names=target_names))
              precision    recall  f1-score   support

     class 0       0.67      1.00      0.80         2
     class 1       0.00      0.00      0.00         1
     class 2       1.00      0.50      0.67         2

    accuracy                           0.60         5
   macro avg       0.56      0.50      0.49         5
weighted avg       0.67      0.60      0.59         5
```

Macroavg: mean of all classes

- Common aggregations: *micro* and *macro* averages. (Tradeoffs?)

Do I have enough labels?

- For training, hundreds to thousands of annotations may be needed for reasonable performance
  - Current work: how to usefully make NLP models with <10 or <100 training examples. "Few-shot learning"
- Exact amounts are difficult to know in advance. Can do a **learning curve** to estimate if more annotations will be useful.

- But where do the labels come from?