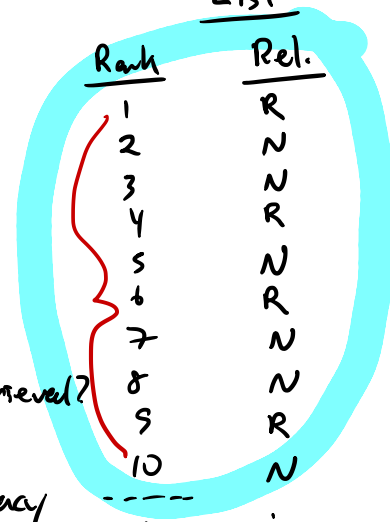
Setup:

- $N = 1{,}000{,}000$ docs in corpus
- $R = 8$ relevant docs
- System retrieves 4 rel. docs in the top 10 results
- What is performance of set retrieved?
- Standard ML measure: accuracy

  acc = frac. of instances correctly predicted

  error = 1 − acc = ... incorrect




Correct?

✓ ✗ ✗ ✓ ✗ ✓ ✗ ✗ ✓ ✗ ✓ ✗ ✓ ✗ ✓ ✗

4 ✓ 6 ✗ 4 ✗ 999986 ✓

$$acc = \frac{\#\checkmark}{N} = \frac{4 + 999986}{1{,}000{,}000} = 99.999\%$$

$$err = \frac{\#\times}{N} = \frac{6+4}{1{,}000{,}000} = \frac{10}{1{,}000{,}000} = 0.001\%$$

Search ensure returns nothing:

$$err = \frac{8}{1{,}000{,}000} = .0008\%$$

How to measure performance w/ massive data imbalance?

⇒ ① Better set-level metrics

② Ranking based metrics

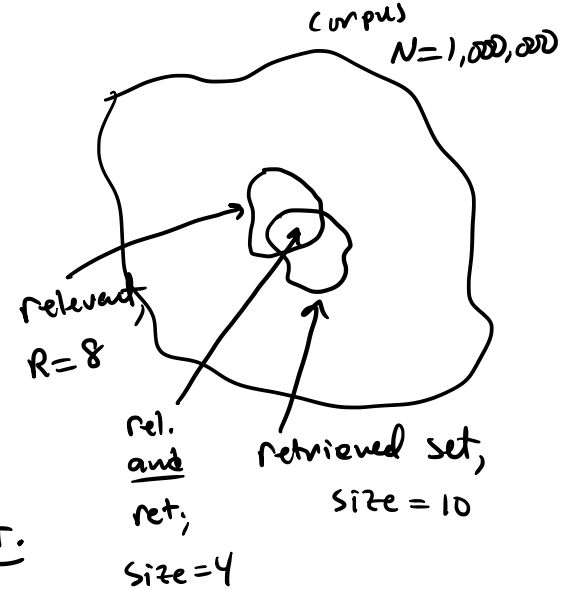
---

- Set-level metrics: precision, recall, F1
- precision = frac. of ret. docs that are rel.

$$= \frac{|\text{rel.} \cap \text{ret.}|}{|\text{ret.}|} = \frac{4}{10} = 0.4$$

- recall = frac. of rel. docs that are ret.

$$= \frac{|\text{rel.} \cap \text{ret.}|}{|\text{rel.}|} = \frac{4}{8} = 0.5$$




- How to combine prec. & recall? maybe just average $= \frac{0.4+0.5}{2} = 0.45$
- But easy to game in IR

① return just the top rel. doc — prec = 1/1 = 1, rec = 1/8 = 0.125; $\frac{p+r}{2} = \frac{1+0.125}{2} = 0.5625$

② return everything — prec = 8/1,000,000 = 0.000008 ≈ 0, rec = 8/8 = 1; $\frac{p+r}{2} \approx \frac{0+1}{2} = 0.5$

How to combine #'s when you want to penalize for any # small?

Digression: Other Kinds of means?

arithmetic mean: $\frac{x_1 + x_2}{2}$ or $\frac{x_1 + x_2 + \cdots + x_n}{n}$ - straight average

geometric mean: $\sqrt{x_1 \cdot x_2}$ or $\sqrt[n]{x_1 \cdot x_2 \cdot x_3 \cdots x_n}$

harmonic mean: $\frac{1}{\left(\frac{1}{x_1} + \frac{1}{x_2}\right)/2} = \frac{2}{\frac{1}{x_1} + \frac{1}{x_2}} = \frac{2x_1x_2}{x_1 + x_2}$ or $\frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$

Thm: h.m. $\leq$ g.m $\leq$ a.m w/equality iff all #'s same

In I.R., we use the h.m., which we call F1

$$F1 = \text{harm. mean}(\text{prec}, \text{rec}) = \frac{2}{\frac{1}{\text{prec}} + \frac{1}{\text{rec}}} = \frac{2 \cdot \text{prec} \cdot \text{rec}}{\text{prec} + \text{rec}}$$

Example

| p | r | a.m. | F1 |
|---|---|---|---|
| 0.4 | 0.5 | 0.45 | 0.444 |
| 1 | 0.125 | 0.5625 | 0.222 |
| 0.00008 | 1 | 0.50004 | 0.00016 |

weighted $F_\beta$

- what about ranked retrieval evaluation?
- ROC curves → used widely in ML & data analysis
  ↳ receiver operator curve
- true positive rate
- false positive rate

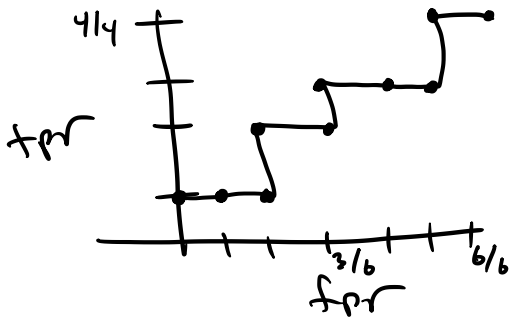
| | | tpr | fpr |
|---|---|---|---|
| 1 | R | 1/4 | 0/6 |
| 2 | N | 1/4 | 1/6 |
| 3 | N | 1/4 | 2/6 |
| 4 | R | 2/4 | 2/6 |
| 5 | N | 2/4 | 3/6 |
| 6 | R | 3/4 | 3/6 |
| 7 | N | 3/4 | 4/6 |
| 8 | N | 3/4 | 5/6 |
| 9 | R | 4/4 | 5/6 |
| 10 | N | 4/4 | 6/6 |

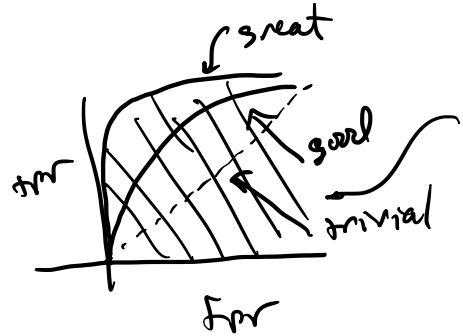
universe of just 10 items

original setup
R=8 N=1M

| tpr | fpr |
|---|---|
| 1/8 | 0/999,992 |
| 2/8 | 1/999,992 |
| 1/8 | 2/999,992 |
| 1/4 | ⋮ |
| ⋮ | ⋮ |

trivial ROC curve




area under curve → AUC

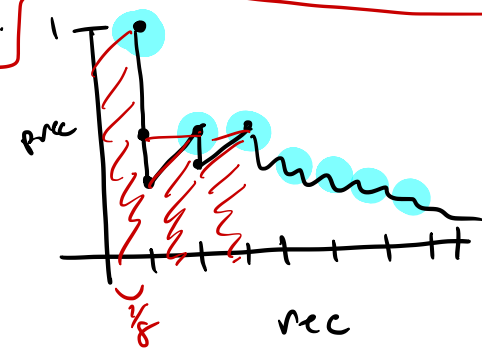
# IR: Ranked Retrieval Metrics

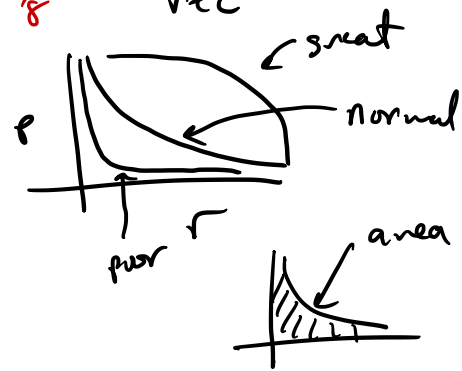
- Prec-rec curves
- avg. prec.
- R-prec.

$N = 1,000,000$

$R = 8$

| Rank | Rel. | prec | rec |
|---|---|---|---|
| 1 | R | 1/1 | 1/8 |
| 2 | N | 1/2 | 1/8 |
| 3 | N | 1/3 | 1/8 |
| 4 | R | 2/4 | 2/8 |
| 5 | N | 2/5 | 2/8 |
| 6 | R | 3/6 | 3/8 |
| 7 | N | 3/7 | 3/8 |
| 8 | N | 3/8 | 3/8 |
| 9 | R | 4/9 | 4/8 |
| 10 | N | 4/10 | 4/8 |
| ⋮ | ⋮ | | |
| | R | ~0 | 5/8 |
| | ⋮ | | |
| | R | ~0 | 6/8 |
| | ⋮ | | |
| | R | ~0 | 7/8 |
| | R | ~0 | 8/8 |




avg. prec. = avg. of prec. at each rel. doc

$$= \frac{1/1 + 2/4 + 3/6 + 4/9 + \cdots 0}{8} = 0.3524$$

R-prec = (prec=rec) = $3/8 = 0.375$