Setting up your goal

Tuesday, September 1, 2020 9:22 PM

① Single number evaluation metric

- Precision: Of the examples recognised true what % is actually true
- Recall: What % of the actually true examples are correctly recognised
- Using 2 metrics causes dilemma of quickly picking a metric to evaluate the model

  ⇒ F_1 score = "Average" of P & R ($\frac{2}{\frac{1}{P}+\frac{1}{R}}$ "Harmonic mean")

- Having a dev set & a single number evaluation metric helps to speed up iterative process
- If we have multiple errors from diff. types of data, take the average

② Satisficing and Optimizing metric

- we might also consider running time of a model to evaluate it.

  ⇒ maximize accuracy subject to running time ≤ 100ms

  Satisficing metric

- Generally for n evaluation metrics available,

  make 1 optimizing & n-1 satisficing metric

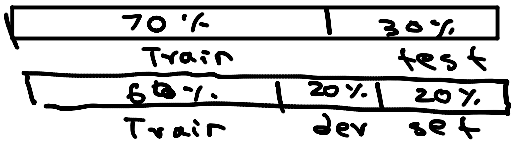
③ Training / Dev / Test distributions

└hold-out cross validation set

- Make sure dev / test sets belong to same distributions
- Choose a dev set and test set to reflect data you expect to get in the future and consider important to do well on.

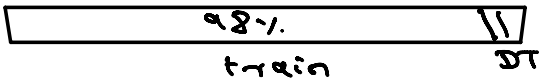
④ Size of dev and test sets

- Old way



reasonable up to 10000 example

- Big data era



millions of examples

- size of test set
  - → set your test set to be big enough to give high confidence in the overall performance of your system
  - → Test set isn't always necessary Dev set is enough but not recommended

⑤ When to change dev/test sets and metrics

- When metrics say one algorithm is better while the company/users prefer another, we should change our metrics

---

Cat dataset examples

Metric + Dev : Prefer A
You/users : Prefer B.

→ Metric: classification error

Algorithm A: 3% error → pornographic

✓ Algorithm B: 5% error

$$\text{Error}: \frac{1}{\sum w^{(i)}} \frac{1}{m_{dev}} \sum_{i=1}^{m_{dev}} w^{(i)} \mathbb{1}\{y_{pred}^{(i)} \neq y^{(i)}\}$$

$y_{pred}^{(i)}$: predicted value (0/1)

→ $w^{(i)} = \begin{cases} 1 & \text{if } x^{(i)} \text{ is non-porn} \\ 10 & \text{if } x^{(i)} \text{ is porn} \end{cases}$

1. So far we've only discussed how to define a metric to evaluate classifiers. ← Place target

2. Worry separately about how to do well on this metric. Aim/shoot at target

$$J = \frac{1}{\sum w^{(i)}} \sum_{i=1}^{m} w^{(i)} \mathcal{L}(\hat{y}^{(i)}, y^{(i)})$$

- when real world data is not as clean as train/test data, even a lower performing model becomes better for deployment
- If doing well on metric dev/test set does not correspond to doing well on application, change metric or the dev/test set