HW3

LING572

Advanced Statistical Methods for NLP

Highlights

- Q1: run the NB learner in MALLET
- Q2: build multi-variate Bernoulli NB learner
- Q3: build multinomial NB learner

Q2

- `build_NB1.sh training_data test_data prior_delta cond_prob_delta model_file sys_output > acc`

- `prior_delta:` delta for calculating $P(c)$

- `cond_prob_delta:` delta for calculating $P(f \mid c)$

Model file

c1 P(c1) logP(c1)   # log is base-10

…

f1 c1 P(f1 | c1) logP(f1 | c1)

f2 c1 P(f2 | c1) logP(f2 | c1)

…

f1 c2 P(f1 | c2) logP(f1 | c2)

f2 c2 P(f2 | c2) logP(f2 | c2)

…

sys_output

instanceName trueClass c1 p1 c2 p2 ...

instanceName will be array:0, array:1, etc.

(c_i, p_i) should be sorted by the value of p_i

$$p_i = P(c_i \mid x) = \frac{P(x \mid c_i)P(c_i)}{P(x)}$$

$$P(x) = \sum_i P(c_i, x) = \sum_i P(x \mid c_i)P(c_i)$$

Underflow Issue

$$p_i = P(c_i | x) = \frac{P(x | c_i)P(c_i)}{P(x)} = \frac{P(x, c_i)}{\sum_{c_i} P(x, c_i)}$$

$$\log P(x, c_1) = -200; \ \log P(x, c_2) = -201; \ \log P(c, x_3) = -202$$

$$p_1 = \frac{10^{-200}}{10^{-200} + 10^{-201} + 10^{-202}} = \frac{1}{1 + 10^{-1} + 10^{-2}} = 100/111 = 0.901$$

$$p_2 = \frac{10^{-1}}{1 + 10^{-1} + 10^{-2}} = 10/111 = 0.09$$

$$p_3 = \frac{10^{-2}}{1 + 10^{-1} + 10^{-2}} = 1/111 = 0.009$$

Efficiency ex. #1

$$\log P(c) \prod_{k=1}^{|V|} P(w_k \mid c)^{N_{ik}} = \log P(c) + \sum_{k=1}^{|V|} \log(P(w_k \mid c)^{N_{ik}}$$

$$= \log P(c) + \sum_{k=1}^{|V|} N_{ik} \log P(w_k \mid c)$$

Efficiency ex. #2

$$P(d_i | c) = P(c)(\prod_{w_k \in d_i} P(w_k | c))(\prod_{w_k \notin d_i} 1 - P(w_k | c))$$

$$= P(c)(\prod_{w_k \in d_i} P(w_k | c))\frac{\prod_{w_k} 1 - P(w_k | c)}{\prod_{w_k \in d_i} 1 - P(w_k | c)}$$

$$= P(c) \prod_{w_k \in d_i} \frac{P(w_k | c)}{1 - P(w_k | c)} \prod_{w_k} 1 - P(w_k | c)$$

Efficiency: ex #3

$$P(c_j | d_i) = \begin{cases} 1 & d_i \text{ has label } c_j \\ 0 & \text{otherwise} \end{cases}$$

$$P(w_t | c_j) = \frac{1 + \sum_{i=1}^{|D|} N_{it} P(c_j | d_i)}{|V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|D|} N_{is} P(c_j | d_i)}$$

Complexity: $O(|V| \cdot |D| \cdot |V| \cdot |C|) = O(|V|^2 \cdot |C| \cdot |D|)$

Efficiency: ex #3

$Z(c_j) = 0$ `for each` c_j

```
for each d_i
  let c_j be the class label of d_i
  for each w_t present in d_i
    let N_it be the number of occurrences of w_t in d_i
    count(w_t, c_j) += N_it
    Z(c_j) += N_it

for each c_j
  for each w_t
```

$$P(w_t \mid c_j) = \frac{1 + \mathrm{count}(w_t, c_j)}{|V| + Z(c_j)}$$

Complexity: $O(|V| \cdot |C| + |D| \cdot \mathrm{avg}(\mathrm{feat/doc}))$

Efficiency: vectorize!

Bad

```
probs = [0.2, 0.2, 0.6]
ent = 0
for prob in probs:
 ent -= prob*math.log2(prob)
```

Good (better)

```
import numpy as np
probs = np.array([0.2, 0.2, 0.6])
ent = -probs*np.log2(probs)
```

(This example only uses element-wise operations;
even more power when doing more bona-fide vector arithmetic.)