Maximum Entropy

Dienstag, 27. April 2021 13:39

> conditional probability $\Longrightarrow$ Maximize conditional likelihood

$p(y \mid x_1, \dots, x_n) \neq p(y, x_1, \dots, x_n)$

↳ not just counts but optimization process

$$p_\lambda(y \mid \mathbf{x}) = \frac{e^{\sum_i \lambda_i \cdot f_i(y, \mathbf{x})}}{\sum_{y' \in Y} e^{\sum_i \lambda_i \cdot f_i(y', \mathbf{x})}} = \frac{1}{Z(x)} \cdot e^{\sum_i \lambda_i \cdot f_i(y, x)}$$

same constant for each y (only depends on x)

→ f_i = feature – 0 or 1 – property of document $\mathbf{x}$ with class y

e.g. occurance of the word "money" + class = spam → 1

λ_i = weight for feature f_i, can be high/positive (= count toward class y), low (= not expressive with class y) or negative (= count against y)

y = class

$\mathbf{x}$ = document vector (e.g. bag-of-words)

LEARNING λ-WEIGHTS

> maximize the conditional log likelihood of the data given the model

$$\max_{\lambda}^{!} \log(p_\lambda(Y \mid X)) = \sum_{(y, \mathbf{x}) \in (Y, X)} \log(p_\lambda(y \mid \mathbf{x}))$$

$$= \sum_{(y, \mathbf{x}) \in (Y, X)} \log\left(\frac{e^{\sum_i \lambda_i f_i(y, \mathbf{x})}}{\sum_{y' \in Y} e^{\sum_i \lambda_i f_i(y', \mathbf{x})}}\right)$$

↳ log + exp cancel out

find derivative of this and optimize = minimize

$$= \sum_{(y, \mathbf{x}) \in (Y, X)} \log\left(\frac{e^{\sum_i \lambda_i f_i(y, \mathbf{x})}}{\sum_{y' \in Y} e^{\sum_i \lambda_i f_i(y', \mathbf{x})}}\right)$$

logarithm + fraction

$$\Leftrightarrow \sum_{(y, \bar{x}) \in (Y, X)} \left(\log \exp \sum_i \lambda_i f_i(y, \bar{x}) - \log \sum_{y' \in Y} \exp \sum_i \lambda_i f_i(y', \bar{x})\right)$$

reformulate against big brackets

$$\Leftrightarrow \underbrace{\sum_{(y, \bar{x}) \in (Y, X)} \log \exp \sum_i \lambda_i f_i(y, \bar{x})}_{\mathcal{A}_\lambda} - \underbrace{\sum_{(y, \bar{x}) \in (Y, X)} \log \sum_{y' \in Y} \exp \sum_i \lambda_i f_i(y', \bar{x})}_{\mathcal{B}_\lambda}$$

$\Rightarrow$ compute derivative of $\mathcal{A}$ and $\mathcal{B}$ separately

$$\frac{\partial \mathcal{A}_\lambda}{\partial \lambda_i} = \sum_{(y, \bar{x}) \in (Y, X)} f_i(y, \bar{x})$$

$\Longrightarrow$ empirical feature count

perfect model: = 1 model predicts the same amounts as are empirically there

$$\frac{\partial \mathcal{B}_\lambda}{\partial \lambda_i} = \sum_{(y, \bar{x}) \in (Y, X)} \sum_{y'} p_\lambda(y' \mid \bar{x}) f_i(y', \bar{x})$$

$\Rightarrow$ predicted feature count

$p_\lambda = 1$ if $y' = \hat{y}$

training instance

⇓

number of correctly switched on training instances minus weighted sum of all switched on instances

d_1 : "Where art thou Romeo"

d_2 : "Where will you go"

	d_1	d_2
where	1	1
art	1	0
thou	1	0
Romeo	1	0
will	0	1
you	0	1
go	0	1

```
class Feature:
    property Label
    property property
    method setLabel
    method setProperty
```

use as boolean to switch weight on or off

if Feature.Label and doc[Feature.property]

```
class MaxEnt:
    property features
    property weight
    method learnFeatures
    method train
```

set

set

list/dict with all features?

same amount, maybe amount of features learned in learnFeatures and weights initialized as a list with same length?

after learning x number of features, initialize list of x number of weights between +2 and −2 ?

?

1. compute probabilities for all instances
2. for every weight$_i$
   count all trainings instances switched on by the property$_i$
   $\partial \mathcal{B} \to$ multiply by repective probability
   $\partial \mathcal{A} \Rightarrow$ of those, count those also switched on by label$_i$

   $\Rightarrow$ $\mathcal{A}_i - \mathcal{B}_i$ → update λ_i ?
3. update λ's

$\Rightarrow$ repeat until convergence

if doc[property$_i$]
  $\partial \mathcal{B}_i$ += 1
  if label(doc) == label$_i$
  $\partial \mathcal{A}_i$ += 1