Machine Learning for Language Processing
ACS 2015/16
Stephen Clark

L3: Maximum Entropy Models

UNIVERSITY OF CAMBRIDGE

Discriminative Models

- Classification requires the class-posterior $P(\omega_j|\boldsymbol{x})$
  - can just directly model the posterior distribution
  - avoids the complexity of modelling the joint distribution $P(\boldsymbol{x}, \omega_j)$

- Form of model called a discriminative model

- Many debates of generative versus discriminative models:
  - discriminative model criterion more closely related to classification process
  - not dependent on generative process being correct
  - joint distribution can be very complicated to accurately model
  - only final posterior distribution needs to be a valid distribution

NER as a Tagging Problem

```
England|I-LOC 's|O fencers|O won|O gold|O on|O
day|I-TIME 4|I-TIME in|O Delhi|I-LOC with|O a|O medal|O
-winning|O performance|O .|O

This|O is|O Prof.|I-PER Black|I-PER 's|O second|O
gold|O of|O the|O Games|O .|O
```

Feature-based Models

- Features encode evidence from the context for a particular tag:

(title caps, `NNP`)	Citibank, Mr.
(suffix `-ing`, `VBG`)	running, cooking
(next word `Inc.`, `I-ORG`)	Lotus Inc.
(previous word `said`, `I-PER`)	said Mr. Vinken

Complex Features

- Features can be arbitrarily complex
  - e.g. document level features
    (document = `cricket` & current word = `Lancashire`, `I-ORG`)
    $\Longrightarrow$ hopefully tag `Lancashire` as `I-ORG` not `I-LOC`

- Features can be combinations of atomic features
  - (current word = `Miss` & next word = `Selfridges`, `I-ORG`)
    $\Longrightarrow$ hopefully tag `Miss` as `I-ORG` not `I-PER`

- Features are not assumed to be (conditionally) independent (given the label)
  – unlike the Naive Bayes classifier

Feature-based Tagging

- How do we incorporate features into a probabilistic tagger?
- Hack the Markov Model tagger to incorporate features
- Maximum Entropy (MaxEnt) Tagging
  - principled way of incorporating features
  - requires sophisticated estimation method

Features in MaxEnt Models

- Features encode elements of the context C useful for predicting tag t

- Features are binary valued functions, e.g.

$$f_i(C,t) = \begin{cases} 1 & \text{if } \texttt{word}(C) = \texttt{Moody} \ \ \& \ \ t = \text{I-ORG} \\ 0 & \text{otherwise} \end{cases}$$

- $\texttt{word}(C) = \texttt{Moody}$ is a *contextual predicate*

- Features determine (`contextual_predicate, tag`) pairs

The Model

$$p(t|C) = \frac{1}{Z(C)} \exp\left(\sum_{i=1}^{n} \lambda_i f_i(C,t)\right)$$

- f_i is a feature
- λ_i is a weight (large value implies informative feature)
- $Z(C)$ is a normalisation constant ensuring a proper probability distribution
- Also known as a *log-linear* model
- Makes no independence assumptions about the features
- Can be used as a general classifer (outside of tagging, e.g. text classification)

Tagging with MaxEnt Models

- The conditional probability of a tag sequence $t_1 \dots t_n$ is

$$p(t_1 \dots t_n | w_1 \dots w_n) \approx \prod_{i=1}^{n} p(t_i | C_i)$$

  given a sentence $w_1 \dots w_n$ and contexts $C_1 \dots C_n$

- The context includes previously assigned tags (for a fixed history)

- Beam search or Viterbi is used to find the most probable sequence (Ratnaparkhi, 1996)

Model Estimation

$$p(t|C) = \frac{1}{Z(C)} \exp\left(\sum_{i=1}^{n} \lambda_i f_i(C,t)\right)$$

- Model estimation involves setting the weight values λ_i
- The model should reflect the data
  $\Longrightarrow$ use the data to *constrain* the model
- What form should the constraints take?
  $\Longrightarrow$ constrain the *expected value* of each feature f_i

The Constraints

$$E_p f_i = \sum_{C,t} p(C,t) f_i(C,t) = K_i$$

- Expected value of each feature must satisfy some constraint K_i
- A natural choice for K_i is the average empirical count:

$$K_i = \quad E_{\tilde{p}} f_i = \frac{1}{N} \sum_{j=1}^{N} f_i(C_j, t_j)$$

derived from the training data $(C_1, t_1), \ldots, (C_N, t_N)$

Choosing the MaxEnt Model

- The constraints do not *uniquely* identify a model
- From those models satisfying the constraints: *choose the Maximum Entropy model*
- Conditional entropy of a model p:

$$H(p) = -\sum_{C,t} \tilde{p}(C)p(t|C)\log p(t|C)$$

The Maximum Entropy Model

- The maximum entropy model is the *most uniform model*
  $\Longrightarrow$ makes no assumptions in addition to what we know from the data

- MaxEnt model is also the *Maximum Likelihood Log-Linear* model

- Set the weights to give the MaxEnt model satisfying the constraints
  $\Longrightarrow$ use *Generalised Iterative Scaling* (GIS)

Generalised Iterative Scaling (GIS)

- Set $\lambda_i^{(0)}$ equal to some arbitrary value (e.g. zero)
- Repeat until convergence:

$$\lambda_i^{(t+1)} = \lambda_i^{(t)} + \frac{1}{C} \log \frac{E_{\tilde{p}} f_i}{E_{p^{(t)}} f_i}$$

where

$$C = \max_{x,y} \sum_{i=1}^{n} f_i(x, y)$$

POS Tagger Features

- The tagger uses binary valued features, e.g.

$$f_i(x, y) = \begin{cases} 1 & \text{if } \texttt{word}(x) = \texttt{the} \ \ \& \ \ y = \text{DT} \\ 0 & \text{otherwise} \end{cases}$$

- $\texttt{word}(\text{x}) = \texttt{the}$ is a *contextual predicate*

- Contextual predicates:

$t_{i-1} = X$	previous tag history
$t_{i-2}t_{i-1} = XY$	previous two tags history
$w_i = X$	current word
$w_{i-1} = X$	previous word
$w_{i-2} = X$	previous previous word
$w_{i+1} = X$	next word
$w_{i+2} = X$	next next word

POS Tagger Features for Rare Words

- These predicates apply to words seen less than 5 times in the data

X is prefix of w_i, $|X| \leq 4$
X is suffix of w_i, $|X| \leq 4$
w_i contains a digit
w_i contains uppercase char
w_i contains a hyphen

- Otherwise the current word predicate applies

Performance

- MaxEnt taggers give close to state-of-the-art accuracy (over 97% on PTB data)
- Training and testing is fast (100s of 1000s of words per second at test time)
- Lots of recent work on tagging other sorts of data, eg tweets
- Recurrent neural networks (probably) give the state-of-the-art for tagging

Contextual Predicates for NER

Condition	Contextual predicate
$f(w_i) < 5$	X is prefix/suffix of w_i, $\lvert X \rvert \leq 4$ w_i contains a digit w_i contains uppercase character w_i contains a hyphen
$\forall w_i$	$w_i = X$ $w_{i-1} = X$, $w_{i-2} = X$ $w_{i+1} = X$, $w_{i+2} = X$
$\forall w_i$	$\mathsf{POS}_i = X$ $\mathsf{POS}_{i-1} = X$, $\mathsf{POS}_{i-2} = X$ $\mathsf{POS}_{i+1} = X$, $\mathsf{POS}_{i+2} = X$
$\forall w_i$	$\mathsf{NE}_{i-1} = X$ $\mathsf{NE}_{i-2}\mathsf{NE}_{i-1} = XY$

Contextual Predicates for NER

Condition	Contextual predicate
$f(w_i) < 5$	w_i contains period
	w_i contains punctuation
	w_i is only digits
	w_i is a number
	w_i is {upper,lower,title,mixed} case
	w_i is alphanumeric
	length of w_i
	w_i has only Roman numerals
	w_i is an initial (X.)
	w_i is an acronym (ABC, A.B.C.)

Contextual Predicates for NER

Condition	Contextual predicate
$\forall w_i$	memory NE tag for w_i unigram tag of w_{i+1} unigram tag of w_{i+2}
$\forall w_i$	w_i in a gazetteer w_{i-1} in a gazetteer w_{i+1} in a gazetteer
$\forall w_i$	w_i not lowercase and $f_{\text{lc}} > f_{\text{uc}}$
$\forall w_i$	unigrams of word type bigrams of word types trigrams of word types

Contextual Predicates for NER

- Moody $\Longrightarrow$ Aa
- A.B.C. $\Longrightarrow$ A.A.A.
- 1,345.00 $\Longrightarrow$ 0,0.0
- Mr. Smith $\Longrightarrow$ Aa. Aa