Machine Learning for natural language processing

Conditional Random Fields

Laura Kallmeyer

Heinrich-Heine-Universität Düsseldorf

Summer 2016

Introduction

- HMM is a generative sequence classifier.
- MaxEnt is a discriminative classifier.
- Today: discriminative sequence classifier, combining HMM and MaxEnt.

Lafferty et al. (2001); Sha & Pereira (2003); Wallach (2002, 2004)

Table of contents

1. Motivation
2. Conditional Random Fields
3. Efficient computation
4. Features

Motivation

- Naive Bayes is a generative classifier assigning a single class to a single input.
- MaxEnt is a discriminative classifier assigning a single class to a single input.
- HMM is a generative sequence classifier assigning sequences of classes to sequences of input symbols.
- CRF is a discriminative sequence classifier assigning sequences of classes to sequences of input symbols.

	single classification	sequence classification
generative	naive Bayes	HMM
discriminative	MaxEnt	CRF

Motivation

- **Generative classifiers** compute the probability of a class y given an input x as

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} = \frac{P(x,y)}{P(x)}$$

- HMM is a generative classifier:

$$P(\mathbf{q}|\mathbf{o}) = \frac{P(\mathbf{o},\mathbf{q})}{P(\mathbf{o})}$$

  For the classification, we have to compute

$$\arg\max_{\mathbf{q}\in Q^n} P(\mathbf{o},\mathbf{q}) = \arg\max_{\mathbf{q}\in Q^n} P(\mathbf{o}|\mathbf{q})P(\mathbf{q})$$

- The computation of the joint probability $P(x,y)$ (here $P(\mathbf{o},\mathbf{q})$) is a complex task.
- In contrast, MaxEnt classifiers directly compute the conditional probability $P(y|x)$ that has to be maximized.

Motivation

The move from generative to discriminative sequence classification (HMM to CRF) has two advantages:

- We do not need to compute the joint probability any longer.
- The strong independence assumptions of HMMs can be relaxed since features in a discriminative approach can capture dependencies that are less local than the n-gram based featues of HMMs.
- Feature weights need not be probabilities, i.e., can have values lower than 0 or greater than 1.

Conditional Random Fields

Goal: determine the best sequence $\mathbf{y} \in C^n$ of classes, given an input sequence $\mathbf{x}$ of length n.

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in C^n} P(\mathbf{y}|\mathbf{x})$$

CRF Applications

Sample applications are:

- POS tagging Ratnaparkhi (1997)
- shallow parsing Sha & Pereira (2003)
- Named Entity Recognition (Stanford NER)[a] Finkel et al. (2005)
- language identification Samih & Maier (2016)

[a] `http://nlp.stanford.edu/software/CRF-NER.shtml`

Conditional Random Fields

The probability of a class sequence for an input sequence depends on features (so-called **potential functions**).

Features refer to the potential class of some input symbol $\mathbf{x}_i$ and to the classes of some other input symbols.

Features are usually indicator functions that will be weighted.

Sample features

$$t(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{x}, i) = \begin{cases} 1 \text{ if } \mathbf{x}_i = \text{“September”} \text{ and } \mathbf{y}_{i-1} = \text{IN} \text{ and } \mathbf{y}_i = \text{NNP} \\ 0 \text{ otherwise} \end{cases}$$

(taken from Wallach (2004))

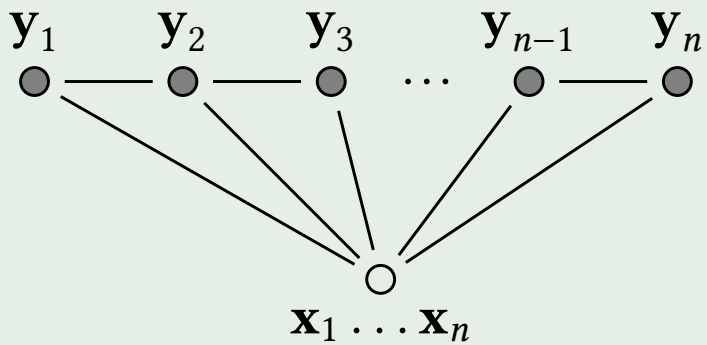
$$s(\mathbf{y}_i, \mathbf{x}, i) = \begin{cases} 1 \text{ if } \mathbf{x}_i = \text{“to”} \text{ and } \mathbf{y}_i = \text{TO} \\ 0 \text{ otherwise} \end{cases}$$

Conditional Random Fields

The dependencies that are expressed within the features can be captured in a graph.

CRF graph

If we have only transition features applying to $\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{x}, i$ and state features applying to $\mathbf{y}_i, \mathbf{x}, i$, we get a chain-structured CRF:



Conditional Random Fields

In order to compute the probability of a class sequence for an input sequence, we

- extract the corresponding features,
- combine them linearly (= multiplying each by a weight and adding them up)
- and then applying a function to this linear combination, exactly as in the MaxEnt case.

In the following, we assume that we have only transition features and state features where the latter can also be considered a transition feature (that gives the same value for all preceding states). I.e., every features has the form $f(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{x}, i)$.

Conditional Random Fields

The f weights for a fixed f but for different input positions receive all the same weight. Therefore we can sum them up before weighting.

From class features to sequence features

$$F(\mathbf{y}, \mathbf{x}) = \sum_{i=1}^{n} f(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{x}, i)$$

Assume that we have features $f_1, \ldots, f_k$, which yield sequence features $F_1, \ldots, F_k$.

We weight these and apply them exactly as in the MaxEnt case:

Conditional class sequence probability

Let λ_j be the weight of features F_j.

$$P(\mathbf{y}|\mathbf{x}) = \frac{\mathrm{e}^{\sum_{i=1}^{k} \lambda_i F_i(\mathbf{y},\mathbf{x})}}{\sum_{\mathbf{y}' \in C^n} \mathrm{e}^{\sum_{i=1}^{k} \lambda_i F_i(\mathbf{y}',\mathbf{x})}}$$

Conditional Random Fields

CRF – POS tagging

Assume that we have POS tags Det, N, Adv, V and features

$$f_1(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{x}, i) = \begin{cases} 1 \text{ if } \mathbf{x}_i = \text{“chief”} \text{ and } \mathbf{y}_{i-1} = \text{Det} \text{ and } \mathbf{y}_i = \text{Adj} \\ 0 \text{ otherwise} \end{cases}$$

$$f_2(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{x}, i) = \begin{cases} 1 \text{ if } \mathbf{x}_i = \text{“chief”} \text{ and } \mathbf{y}_i = \text{N} \\ 0 \text{ otherwise} \end{cases}$$

$$f_3(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{x}, i) = \begin{cases} 1 \text{ if } \mathbf{x}_i = \text{“talks”} \text{ and } \mathbf{y}_{i-1} = \text{Det} \text{ and } \mathbf{y}_i = \text{N} \\ 0 \text{ otherwise} \end{cases}$$

$$f_4(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{x}, i) = \begin{cases} 1 \text{ if } \mathbf{x}_i = \text{“talks”} \text{ and } \mathbf{y}_{i-1} = \text{Adj} \text{ and } \mathbf{y}_i = \text{N} \\ 0 \text{ otherwise} \end{cases}$$

$$f_5(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{x}, i) = \begin{cases} 1 \text{ if } \mathbf{x}_i = \text{“talks”} \text{ and } \mathbf{y}_{i-1} = \text{N} \text{ and } \mathbf{y}_i = \text{V} \\ 0 \text{ otherwise} \end{cases}$$

$$f_6(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{x}, i) = \begin{cases} 1 \text{ if } \mathbf{x}_i = \text{“the”} \text{ and } \mathbf{y}_i = \text{Det} \\ 0 \text{ otherwise} \end{cases}$$

Weights: $\lambda_1 = 2$, $\lambda_2 = 5$, $\lambda_3 = 9$, $\lambda_4 = 8$, $\lambda_5 = 7$, $\lambda_6 = 20$.

Conditional Random Fields

CRF – POS tagging

Assume that we have a sequence "the chief talks". Which of the following probabilities is higher? $P(Det\ N\ V \mid \text{the chief talks})$ or $P(Det\ Adj\ N \mid \text{the chief talks})$?

Weighted feature sums for both:

1. *Det N V*: $20 + 5 + 7 = 32$
2. *Det Adj N*: $20 + 2 + 8 = 30$

Consequently, *Det N V* has a slightly higher probability.

(In real applications, we have of course many more features.)

Efficient computation

In the following, we assume chain-structured CRF (see example). Let $\mathbf{x}$ be our input sequence of length n. We have features $f_1, \ldots f_k$. Each label sequence is augmented with an initial `start` and a final `end`. Let C be the set of class labels.

We define a set of $C \times C$ matrices $M_1(\mathbf{x}), M_2(\mathbf{x}), \ldots M_{n+1}(\mathbf{x})$ where for all i, $1 \leq i \leq n+1$ and all classes c, c':

$$M_i(c, c') = \mathrm{e}^{\sum_{j=1}^{k} \lambda_j f_j(c, c', \mathbf{x}, i)}$$

With this, we can compute

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=1}^{n+1} M_i(\mathbf{y}_{i-1}, \mathbf{y}_i)$$

Efficient computation

In order to obtain the probability, we have to compute
$Z = \sum_{\mathbf{y}' \in C^n} \mathrm{e}^{\sum_{j=1}^{k} \lambda_j F_j(\mathbf{y}', \mathbf{x})}$

As in the HMM case, we can use the forward-backward algorithm in order to compute this efficiently.

We compute $\alpha_i(c) = \sum_{\mathbf{y}' \in C^{i-1}} \mathrm{e}^{\sum_{j=1}^{k} \lambda_j F_j(\mathbf{y}', \mathbf{x})}$ in a way similar to the HMM foward computation:

Forward computation

1. $\alpha_0(c) = 1$ if $c = \texttt{start}$, else $\alpha_0(c) = 0$.
2. $\alpha_i(c) = \sum_{c' \in C} \alpha_{i-1}(c') M_i(c', c)$ for $1 \leq i \leq n$
3. $Z = \sum_{c \in C} \alpha_n(c)$

Efficient computation

Probability calculation in CRF

$C = \{A, B\}$, $\mathbf{x} = abb$. Features $f_{c,c',x}$ and their weights:

$f_{s,A,a} : 2$ $\quad f_{A,A,b} : 1$ $\quad f_{B,A,b} : 0.3$

$f_{s,B,a} : 0.5$ $\quad f_{A,B,b} : 2$ $\quad f_{B,B,b} : 4$

start	1			
A	0	e^2	$e^2e^1 + e^{0.5}e^{0.3} = 22.31$	$22.31e^1 + 144.62e^{0.3}$
B	0	$e^{0.5}$	$e^2e^2 + e^{0.5}e^4 = 144.62$	$22.31e^2 + 144.62e^4$
	0	1	2	3

$Z = 255.86 + 8060.83 = 8316.69$

$$P(AAA|abb) = \frac{e^{2+1+1}}{8316.69} = 0.0066 \qquad P(BBB|abb) = \frac{e^{0.5+4+4}}{8316.69} = 0.59$$

$$P(ABB|abb) = \frac{e^{2+2+4}}{8316.69} = 0.39 \qquad P(BAB|abb) = \frac{e^{0.5+0.3+2}}{8316.69} = 0.002$$

$$P(BBA|abb) = \frac{e^{0.5+4+0.3}}{8316.69} = 0.015 \qquad P(AAB|abb) = \frac{e^{2+1+2}}{8316.69} = 0.018$$

$$P(ABA|abb) = \frac{e^{2+2+0.3}}{8316.69} = 0.009 \qquad P(BAA|abb) = \frac{e^{0.5+0.3+1}}{8316.69} = 0.0007$$

Efficient computation

In order to obtain the best class sequence, we can use the viterbi algorithm:

Viterbi for CRF

1. $v_0(c) = 1$ if $c = \texttt{start}$, else $v_0(c) = 0$.
2. $v_i(c) = \max_{c' \in C}(v_{i-1}(c')M_i(c', c))$ for $1 \leq i \leq n$

If we keep additional backpointers to the c' that has lead to the maximal value, we can read off the best class sequence, starting from the maximal value $v_n(c)$ we have for any of the classes c. If this best class for n is c, the probability of the best class sequence is

$$\frac{v_n(c)}{Z} = \frac{v_n(c)}{\sum_{c \in C} \alpha_n(c)}$$

Efficient computation

Classification in CRF

$C = \{A, B\}$, $\mathbf{x} = abb$. Features $f_{c,c',x}$ and their weights:

$f_{s,A,a} : 2$ $\quad f_{A,A,b} : 1$ $\quad f_{B,A,b} : 0.3$

$f_{s,B,a} : 0.5$ $\quad f_{A,B,b} : 2$ $\quad f_{B,B,b} : 4$

`start`	1			
A	0	e^2, `start`	e^2e^1, A	$e^{4.5}e^{0.3}$, B
B	0	$e^{0.5}$, `start`	$e^{0.5}e^4$, B	$e^{4.5}e^4$, B
	0	1	2	3

Best class sequence: BBB, probability $\frac{e^{8.5}}{8316.69} = 0.59$

Features

In general, we can have any type of features depending on the entire input sequence, the position i, and the classes of $\mathbf{x}_i$ and $\mathbf{x}_{i-1}$.

Relaxed independence assumptions compared to HMM.

Shallow parsing Sha & Pereira (2003)

Shallow parsing: identify chunks (= non-recursive noun phrases) without analyzing their internal structure.

The chunker assigns a chunk label B, I or O (begin of chunk, inside chunk, outside chunk) to each word.

The CRF classifier assigns a pair of chunk labels to a word $\mathbf{x}_i$, namely the concatenation of the chunk labels of $\mathbf{x}_{i-1}$ and of $\mathbf{x}_i$.

All features $f(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{x}, i)$ are indicator functions, indicating that

- some predicate $p(\mathbf{x}, i)$ holds and
- some predicate $q(\mathbf{y}_{i-1}, \mathbf{y}_i)$ holds.

Features

Shallow parsing Sha & Pereira (2003)

- Sample predicates $p(\mathbf{x}, i)$:
  - $\mathbf{x}_{i+2} = w$,
  - $\mathbf{x}_{i-1} = w, \mathbf{x}_i = w'$,
  - $POS(\mathbf{x}_{i-1}) = t, POS(\mathbf{x}_i) = t'$,
  - ...
- Sample predicates $q(\mathbf{y}_{i-1}, \mathbf{y}_i)$:
  - $\mathbf{y}_i = cc'$ (c, c' are the labels assigned by the chunker),
  - $\mathbf{y}_{i-1} = cc', \mathbf{y}_i = c'c''$,
  - $\mathbf{y}_i = xc'$ where x can be any label,
  - ...

(w, w' are specific words, t, t' specific POS tags and c, c', c'' specific labels from $\{B, I, O\}$.)

In total, Sha & Pereira (2003) use 3,8 million features.

References

Finkel, Jenny Rose, Trond Grenager & Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics* ACL '05, 363–370. Stroudsburg, PA, USA: Association for Computational Linguistics. doi:10.3115/1219840.1219885. http://dx.doi.org/10.3115/1219840.1219885.

Lafferty, John, Andrew McCallum & Fernando Pereira. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *International conference on machine learning*, .

Ratnaparkhi, Adwait. 1997. A simple introduction to maximum entropy models for natural language processing. Tech. Rep. 97–08 Institue for Research in Cognitive Science, University of Pennsylvania.

Samih, Younes & Wolfgang Maier. 2016. Detecting code-switching in moroccan arabic. In *Proceedings of SocialNLP @ IJCAI-2016*, New York. To appear.

Sha, Fei & Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *Proceedings of human language technology, naacl*, .

Wallach, Hana M. 2002. *Efficient training of conditional random fields*: University of Edinburgh dissertation.

Wallach, Hana M. 2004. Conditional random fiels: An introduction. Tech. rep. University of Pennsylvania. Technical Report (CIS), Paper 22.