Log-Linear Models with Structured Outputs

Introduction to Natural Language Processing
Computer Science 585—Fall 2009
University of Massachusetts Amherst

David Smith
including slides from Andrew McCallum

Overview

- Sequence labeling task (cf. POS tagging)
- Independent classifiers
- HMMs
- (Conditional) Maximum Entropy Markov Models
- Conditional Random Fields
- Beyond Sequence Labeling

Sequence Labeling

- Inputs: $x = (x_1, \ldots, x_n)$
- Labels: $y = (y_1, \ldots, y_n)$
- Typical goal: Given x, predict y

- Example sequence labeling tasks
  - Part-of-speech tagging
  - Named-entity-recognition (NER)
    - Label people, places, organizations

NER Example:

Red Sox and Their Fans Let Loose

Elise Amendola/Associated Press

Fans of the slugger David Ortiz in Boston's Copley Square.

By PETE THAMEL
Published: October 31, 2007

BOSTON, Oct. 30 — Jonathan Papelbon turned Boston's World Series victory parade into a full-scale dance party Tuesday as the Red Sox put an exclamation point on the 2007 season.

E-MAIL
PRINT
REPRINTS
SAVE

First Solution: Maximum Entropy Classifier

- Conditional model $p(y|x)$.
  - Do not waste effort modeling $p(x)$, since x is given at test time anyway.
  - Allows more complicated input features, since we do not need to model dependencies between them.
- Feature functions $f(x,y)$:
  - $f_1(x,y)$ = { word is Boston & y=Location }
  - $f_2(x,y)$ = { first letter capitalized & y=Name }
  - $f_3(x,y)$ = { x is an HTML link & y=Location}

First Solution: MaxEnt Classifier

- How should we choose a classifier?

- Principle of maximum entropy
  - We want a classifier that:
    - Matches feature constraints from training data.
    - Predictions maximize entropy.

- There is a unique, exponential family distribution that meets these criteria.

First Solution: MaxEnt Classifier

- $p(y|x;\theta)$, inference, learning, and gradient.
- (ON BOARD)

First Solution: MaxEnt Classifier

- Problem with using a maximum entropy classifier for sequence labeling:
- It makes decisions at each position independently!

Second Solution: HMM

$$P(\mathbf{y},\mathbf{x}) = \prod_t P(y_t \mid y_{t-1}) P(x \mid y_t)$$

- Defines a generative process.
- Can be viewed as a weighted finite state machine.

Second Solution: HMM

- HMM problems: (ON BOARD)
  - Probability of an input sequence.
  - Most likely label sequence given an input sequence.
  - Learning with known label sequences.
  - Learning with unknown label sequences?

Second Solution: HMM

- How can represent we multiple features in an HMM?
  - Treat them as conditionally independent given the class label?
    - The example features we talked about are not independent.
  - Try to model a more complex generative process of the input features?
    - We may lose tractability (i.e. lose a dynamic programming for exact inference).

Second Solution: HMM

- Let’s use a conditional model instead.

Third Solution: MEMM

- Use a series of maximum entropy classifiers that know the previous label.
- Define a Viterbi algorithm for inference.

$$P(\mathbf{y} \mid \mathbf{x}) = \prod_t P_{y_{t-1}}(y_t \mid \mathbf{x})$$

Third Solution: MEMM

- Finding the most likely label sequence given an input sequence and learning.
- (ON BOARD)

Third Solution: MEMM

- Combines the advantages of maximum entropy and HMM!
- But there is a problem…

Problem with MEMMs: Label Bias

- In some state space configurations, MEMMs essentially completely ignore the inputs.

- Example (ON BOARD).

- This is not a problem for HMMs, because the input sequence is generated by the model.

Fourth Solution: Conditional Random Field

- Conditionally-trained, undirected graphical model.
- For a standard linear-chain structure:

$$P(\mathbf{y} \mid \mathbf{x}) = \prod_t \Psi_k(y_t, y_{t-1}, \mathbf{x})$$

$$\Psi_k(y_t, y_{t-1}, \mathbf{x}) = \exp\left(\sum_k \lambda_k f(y_t, y_{t-1}, \mathbf{x}) \right)$$

Fourth Solution: CRF

- Finding the most likely label sequence given an input sequence and learning. (ON BOARD)

Fourth Solution: CRF

- Have the advantages of MEMMs, but avoid the label bias problem.

- CRFs are globally normalized, whereas MEMMs are locally normalized.

- Widely used and applied. CRFs give state-the-art results in many domains.

Example Applications

- CRFs have been applied to:
  - Part-of-speech tagging
  - Named-entity-recognition
  - Table extraction
  - Gene prediction
  - Chinese word segmentation
  - Extracting information from research papers.
  - Many more…