Computational Linguistics 1

CMSC/LING 723, LBSC 744

Kristy Hollingshead Seitz
Institute for Advanced Computer Studies
University of Maryland

Lecture 10: 4 October 2011

Agenda

- HW1 – graded by Thursday
- HW2 – graded by next Tuesday
- HW3 – due next Thursday 10/13
- Questions, comments, concerns?
- Unsupervised Learning "Sneak Peek"
- Tagging Tasks

Agenda

- HW
- Unsupervised Learning "Sneak Peek"
- Tagging Tasks

HMMs: Three Problems

- **Likelihood:** Given an HMM $\lambda = (A, B, \prod)$, and a sequence of observed events O, find $P(O|\lambda)$
- **Decoding:** Given an HMM $\lambda = (A, B, \prod)$, and an observation sequence O, find the most likely (hidden) state sequence
- **Learning:** Given a set of observation sequences and the set of states Q in λ, compute the parameters A and B

Supervised Training

- Transition Probabilities
  - Any $P(t_i \mid t_{i-1}) = C(t_{i-1}, t_i) / C(t_{i-1})$, from the tagged data
  - Example: for P(NN|VB), count how many times a noun follows a verb and divide by the total number of times you see a verb
- Emission Probabilities
  - Any $P(w_i \mid t_i) = C(w_i, t_i) / C(t_i)$, from the tagged data
  - For P(bank|NN), count how many times bank is tagged as a noun and divide by how many times anything is tagged as a noun
- Priors
  - Any $P(q_1 = t_i) = \pi_i = C(t_i)/N$, from the tagged data
  - For π_{NN} , count the number of times NN occurs and divide by the total number of tags (states)
  - A better way?

Unsupervised Training

- No labeled/tagged training data
- No way to compute MLEs directly
- How do we deal?
  - Make an initial guess for parameter values
  - Use this guess to get a better estimate
  - Iteratively improve the estimate until some convergence criterion is met

Expectation Maximization (EM)

Motivating Example

- Let observed events be the grades given out in, say, CMSC723
- Assume grades are generated by a probabilistic model described by single parameter μ
  - P(A) = 1/2, P(B) = μ, P(C) = 2 μ, P(D) = 1/2 - 3 μ
  - Number of 'A's observed = 'a', 'b' number of 'B's, etc.
- Compute MLE of μ given 'a', 'b', 'c' and 'd'

Adapted from Andrew Moore's Slides
http://www.autonlab.org/tutorials/gmm.html

Motivating Example

- Recall the definition of MLE:
  ".... maximizes likelihood of data given the model."
- Okay, so what's the likelihood of data given the model?
  - $P(Data|Model) = P(a,b,c,d|\mu) = (1/2)^a(\mu)^b(2\mu)^c(1/2-3\mu)^d$
  - L = log-likelihood = $\log P(a,b,c,d|\mu)$
    $= a \log(1/2) + b \log \mu + c \log 2\mu + d \log(1/2-3\mu)$
- How to maximize L w.r.t μ ? [Think Calculus]
  - $\delta L/\delta\mu = 0$; $(b/\mu) + (2c/2\mu) - (3d/(1/2-3\mu)) = 0$
  - $\mu = (b+c)/6(b+c+d)$
- We got our answer without EM. Boring!

Motivating Example

- Now suppose:
  - P(A) = 1/2, P(B) = μ, P(C) = 2 μ, P(D) = 1/2 - 3 μ
  - Number of 'A's and 'B's = h, c 'C's, and d 'D's
- Part of the observable information is hidden
- Can we compute the MLE for μ now?
- Chicken and egg:
  - If we knew 'b' (and hence 'a'), we could compute the MLE for μ
  - But we need μ to know how the model generates 'a' and 'b'
- Circular enough for you?

The EM Algorithm

- Start with an initial guess for μ (μ_0)
- t = 1; Repeat:
  - $b_t = \mu_{(t-1)}h/(1/2 + \mu_{(t-1)})$
    [**E-step:** Compute expected value of b given μ]
  - $\mu_t = (b_t + c)/6(b_t + c + d)$
    [**M-step:** Compute MLE of μ given b]
  - t = t + 1
- Until some convergence criterion is met

The EM Algorithm

- Algorithm to compute MLEs for model parameters when information is hidden
- Iterate between Expectation (E-step) and Maximization (M-step)
- Each iteration is guaranteed to increase the log-likelihood of the data (improve the estimate)
- Good news: It will always converge to a maximum
- Bad news: It will always converge to a maximum

Applying EM to HMMs

- Just the intuition…more details next week, gory details in CMSC 773
- The problem:
  - State sequence is unknown
  - Estimate model parameters: A, B & ∏
- Introduce two new observation statistics:
  - Number of transitions from q_i to q_j (ξ)
  - Number of times in state q_i (ϒ)
- The EM algorithm can now be applied

Applying EM to HMMs

- Start with initial guesses for A, B and Π
- t = 1; Repeat:
  - E-step: Compute expected values of ξ, Υ using A_t, B_t, Π_t
  - M-step: Compute MLE of A, B and Π using ξ_t, Υ_t
  - t = t + 1
- Until some convergence criterion is met
- Produces an HMM model (A, B and Π) without the need for tagged training data

Agenda

- HW
- Unsupervised Learning "Sneak Peek"
- Tagging Tasks

Part of Speech (POS) Tagging

Allen Iverson is an inconsistent player. While he can shoot very well, some nights he will score only a few points.

⇩

(NNP Allen) (NNP Iverson) (VBZ is) (DT an) (JJ inconsistent) (NN player) (. .) (IN While) (PRP he) **(MD can)** (VB shoot) (RB very) **(RB well)** (, ,) (DT some) (NNS nights) (PRP he) **(MD will)** **(VB score)** (RB only) (DT a) (JJ few) **(NNS points)** (. .)

Tagging tasks

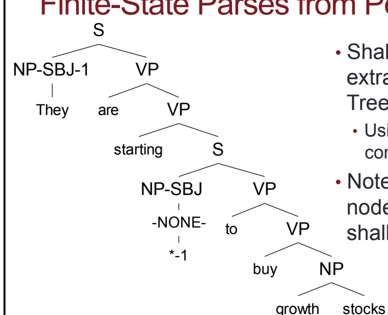
- Start with a sentence: "They are starting to buy growth stocks"
- Identify...
  - Parts of speech?
  - Noun phrases?
  - Verb phrases?
  - Named entities?
  - Co-reference resolution...?

Finite-State "Parsing"

[NP They]	[VP are	starting	to	buy]	[NP growth	stocks]
B-NP	B-VP	I-VP	I-VP	I-VP	B-NP	I-NP

- Flat sequences of base phrases
  - No embedding
- Representation as tag sequences rather than brackets
  - Allows for finite-state processing
  - Referred to as "BIO" tagging
- CoNLL-2000 Shared Task: Chunking
  - An extension of NP-Chunking

Finite-State Parses from Penn Treebank



- Shallow constituents extracted from Penn Treebank trees
  - Using publicly-available conversion script
- Note: sequential VP nodes become one shallow constituent

[NP They] [VP are starting to buy] [NP growth stocks]

Other shallow parsing–like tasks

- **Shallow parsing** (or "chunking") uses 11 different node-labels
  - (NP the boy) (VP saw) (NP his brother)
- **NP chunking** only annotates for noun-phrases
  - (NP the boy) saw (NP his brother)
  - B-NP/the I-NP/boy O/saw B-NP/his I-NP/brother
- **Base-phrase parsing** extracts only those phrases at the "bottom" of the full-parse tree (nodes with only-POS children)

Phrase Tagging

- Named Entity Recognition (NER)
  (persons, organizations, geographical locations, misc)

After receiving his M.B.A. from Harvard Business School, Richard F. America accepted a faculty position at the McDonough School of Business (Georgetown University) in Washington.

⇩

After receiving his **[MISC M.B.A.]** from **[ORG Harvard Business School]**, **[PER Richard F. America]** accepted a faculty position at the **[ORG McDonough School of Business]** (**[ORG Georgetown University]**) in **[LOC Washington]**.

NER Issues

- Named entity phrases are a subset of NPs
  - We can find NPs, so label only NPs
- CoNLL03 shared task
- NE phrases could be embedded
  - How to resolve embeddings?
  - Avoid embedding – 'enlarge' NE phrases

Named Entity Extraction

- NP chunking is shallow parsing with only NP categories
- Named entity extraction is an NP chunking style application that brackets and labels instances of named entities
  - (CO Microsoft) chairman (PER Bill Gates) of (LOC Redmond, WA) . . . where '(CO' denotes a company, '(PER' a person and '(LOC' a location
- One might imagine hierarchical structures, though shallow such as the above is more common
- Bio-informatics applications use such techniques for gene name extraction
- Effective features include capitalization patterns and lists of common names
- Finite state approaches are quite effective

Other Example Tasks

- Morphological analysis
- Segmentation
  - Chinese word-segmentation
- Supertagging
- Word Sense Disambiguation
- Sentence Coherence
- Preposition Identification
- Question Classification
- Spam Filtering

  :

  :

Segmentation as a Tagging Task

- Morphological analysis
  - Dividing a word into its root and stem(s)
  - jump, jumped, jumping ⇒
    (R jump), (R jump) (ST ed), (R jump) (ST ing)
- Chinese word segmentation
  - Chinese text doesn't separate "words" with whitespace as in English text
  - So the sentence: "the boy saw his brother" would be "theboysawhisbrother"
  - Segmentation is the process of inserting spaces
  - Ambiguity in multiple reasonable segmentations

Supertagging as a Tagging Task

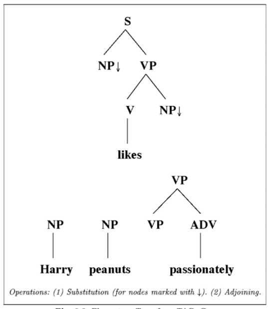


Fig. 8.2. Elementary Trees for a TAG, G_1.

Supertagging as a Tagging Task

- Treating elementary trees as POS-tags called 'Supertagging'
- Large ambiguities in elementary trees for word
  - Much worse than POS-tag ambiguity
  - Issues like subcategorization
- POS-tagging approaches reach low 90s in accuracy
- Has been called 'almost parsing'

Uses of these finite-state/tagging models

- Pruning in multi-pass parsing strategies
  - Supertagging with the XTAG system
  - NP Chunking for the Ratnaparkhi parser
- Providing features for other models
  - Statistical machine translation
- Class-based language modeling
  - Substantial recent improvements with supertagging approach by Wen Wang and Mary Harper

Agenda: Summary

- Unsupervised Learning "Sneak Peek"
- Tagging Tasks
- Take a look at HW3!