Recall: Multiclass Classification

- Logistic regression: \( P(y|x) = \frac{\exp(w^T f(x,y))}{\sum_{y'\in\mathcal{Y}} \exp(w^T f(x,y'))} \)

  Gradient (unregularized):
  \[
  \frac{\partial}{\partial w_i} \mathcal{L}(x_j, y_j^*) = f_i(x_j, y_j^*) - \mathbb{E}_y[f_i(x_j, y)]
  \]

- SVM: defined by quadratic program (minimization, so gradients are flipped)

  Loss-augmented decode
  \[
  \xi_j = \max_{y\in\mathcal{Y}} w^T f(x_j, y) + \ell(y, y_j^*) - w^T f(x_j, y_j^*)
  \]

  Subgradient (unregularized) on \( j \)th example = \( f_i(x_j, y_{\text{max}}) - f_i(x_j, y_j^*) \)

Recall: Optimization

- Stochastic gradient *ascent*
  \[
  w \leftarrow w + \alpha g, \quad g = \frac{\partial}{\partial w} \mathcal{L}
  \]

  Adagrad:
  \[
  w_i \leftarrow w_i + \alpha \frac{1}{\sqrt{\epsilon + \sum_{t=1}^{T} g_{t,i}^2}} g_{t,i}
  \]

- SGD/AdaGrad have a batch size parameter

  Large batches (>50 examples): can parallelize within batch

  ...but bigger batches often means more epochs required because you make fewer parameter updates

  Shuffling: online methods are sensitive to dataset order, shuffling helps!

Administrivia

- Mini 1 due today
- Project 1 out today, due September 27
  - Viterbi algorithm, CRF NER system, extension
  - Extension should be substantial: don’t just try one additional feature (see syllabus/spec for discussion, samples on website)
  - This class will cover what you need to get started on it, the next class will cover everything you need to complete it
This Lecture

- Sequence modeling
- HMMs for POS tagging
- HMM parameter estimation
- Viterbi, forward-backward

Linguistic Structures

- Language is tree-structured
- Understanding syntax fundamentally requires trees — the sentences have the same shallow analysis

POS Tagging

- What tags are out there?

_Ghana’s ambassador should have set up the big meeting in DC yesterday_.

Tanenhaus et al. (1995)
POS Tagging

Open class (lexical) words

Nouns
- Proper: IBM, Italy
- Common: cat, cats, snow

Verbs
- Main: see, registered

Adjectives
- Yellow

Adverbs
- Slowly

Numbers
- 122,312, one

Auxiliary
- Can, had

Prepositions
- To, with

Particles
- Off, up

Closed class (functional)

Determiners
- The, some

Conjunctions
- And, or

Pronouns
- He, its

POS Tagging

Fed raises interest rates 0.5 percent

VBD: Fed
VBN: raises
VB: interest
VBP: rates
VBZ: 0.5
NNS: percent

I’m 0.5% interested in the Fed’s raises!

‣ Other paths are also plausible but even more semantically weird...
‣ What governs the correct choice? Word + context
  ▶ Word identity: most words have <=2 tags, many have one (percent, the)
  ▶ Context: nouns start sentences, nouns follow verbs, etc.

What is this good for?

‣ Text-to-speech: record, lead
‣ Preprocessing step for syntactic parsers
‣ Domain-independent disambiguation for other tasks
‣ (Very) shallow information extraction

Sequence Models

Input \( x = (x_1, \ldots, x_n) \)  
Output \( y = (y_1, \ldots, y_n) \)

‣ POS tagging: \( x \) is a sequence of words, \( y \) is a sequence of tags
‣ Today: generative models \( P(x, y) \); discriminative models next time
Hidden Markov Models

- Input $x = (x_1, ..., x_n)$ Output $y = (y_1, ..., y_n)$
- Model the sequence of $y$ as a Markov process (dynamics model)
- Markov property: future is conditionally independent of the past given the present
  \[ P(y_3|y_1, y_2) = P(y_3|y_2) \]
- Lots of mathematical theory about how Markov chains behave
- If $y$ are tags, this roughly corresponds to assuming that the next tag only depends on the current tag, not anything before

Observation $(x)$ depends only on current state $(y)$
- Multinomials: tag x tag transitions, tag x word emissions
- $P(x|y)$ is a distribution over all words in the vocabulary — not a distribution over features (but could be!)

Transitions in POS Tagging

- Dynamics model $P(y_1) \prod_{i=2}^{n} P(y_i|y_{i-1})$
  - VBD VB
  - VBN VBZ VBP VBZ
  - NNP NNS NN NNS CD NN
  - Fed raises interest rates 0.5 percent

- $P(y_1 = NNP)$ likely because start of sentence
- $P(y_2 = VBZ|y_1 = NNP)$ likely because verb often follows noun
- $P(y_3 = NN|y_2 = VBZ)$ direct object follows verb, other verb rarely follows past tense verb (main verbs can follow modals though!)

Estimating Transitions

- $P(tag \mid NN) = (0.5 \cdot 0.5 NNS)$
- How to smooth?
  - One method: smooth with unigram distribution over tags
  \[ P(tag|tag_{-1}) = (1 - \lambda) \hat{P}(tag|tag_{-1}) + \lambda \hat{P}(tag) \]
  \[ \hat{P} = \text{empirical distribution (read off from data)} \]
Emissions in POS Tagging

- Emissions $P(x | y)$ capture the distribution of words occurring with a given tag
- $P(\text{word} | \text{NN}) = (0.05 \text{ person}, 0.04 \text{ official}, 0.03 \text{ interest}, 0.03 \text{ percent} ...)$
- When you compute the posterior for a given word’s tags, the distribution favors tags that are more likely to generate that word
- How should we smooth this?

Inference in HMMs

- Input $x = (x_1, ..., x_n)$  Output $y = (y_1, ..., y_n)$
- $P(y, x) = P(y_1) \prod_{i=2}^{n} P(y_i | y_{i-1}) \prod_{i=1}^{n} P(x_i | y_i)$
- Inference problem: $\text{argmax}_y P(y|x) = \text{argmax}_y \frac{P(y, x)}{P(x)}$
- Exponentially many possible $y$ here!
- Solution: dynamic programming (possible because of Markov structure!)
- Many neural sequence models depend on entire previous tag sequence, need to use approximations like beam search
Viterbi Algorithm

- Best (partial) score for a sequence ending in state $s$:
  $\text{score}_1(s) = P(s)P(x_1 | s)$

- "Think about" all possible immediate prior state values. Everything before that has already been accounted for by earlier stages.

Slide credit: Vivek Srikumar
Forward-Backward Algorithm

- In addition to finding the best path, we may want to compute marginal probabilities of paths \( P(y_i = s | x) \)

\[
P(y_i = s | x) = \sum_{y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_n} P(y_i | x)
\]

- What did Viterbi compute? \( P(y_{\text{max}} | x) = \max_{y_1, \ldots, y_n} P(y | x) \)

- Can compute marginals with dynamic programming as well using an algorithm called forward-backward

1. **Initial:** For each state \( s \), calculate

\[
score_1(s) = P(s)P(x_1 | s) = \pi_s B_{x_1,s}
\]

2. **Recurrence:** For \( i = 2 \) to \( n \), for every state \( s \), calculate

\[
score_i(s) = \max_{y_{i-1}} P(s | y_{i-1})P(x_i | s)score_{i-1}(y_{i-1})
= \max_{y_{i-1}} A_{y_{i-1},s} B_{s,x_i}score_{i-1}(y_{i-1})
\]

3. **Final state:** calculate

\[
\max_{y_n} P(y, x | \pi, A, B) = \max_{s} score_n(s)
\]

This only calculates the max. To get final answer (argmax),
- keep track of which state corresponds to the max at each step
- build the answer using these back pointers

slide credit: Vivek Srikumar
\[
\begin{align*}
P(y_3 = 2 | x) &= \text{sum of all paths through state 2 at time 3} \\
&= \text{sum of all paths}
\end{align*}
\]

- Initial: \( \alpha_1(s) = P(s)P(x_1|s) \)
- Recurrence: \( \alpha_t(s_t) = \sum_{s_{t-1}} \alpha_{t-1}(s_{t-1})P(s_t|s_{t-1})P(x_t|s_t) \)
- Same as Viterbi but summing instead of maxing!
- These quantities get very small!
  Store everything as log probabilities

\[
\begin{align*}
P(y_3 = 2 | x) &= \text{sum of all paths through state 2 at time 3} \\
&= \text{sum of all paths}
\end{align*}
\]

- Easiest and most flexible to do one pass to compute and one to compute

- Initial: \( \beta_{n}(s) = 1 \)
- Recurrence: \( \beta_{t}(s_t) = \sum_{s_{t+1}} \beta_{t+1}(s_{t+1})P(s_{t+1}|s_{t})P(x_{t+1}|s_{t+1}) \)
- Big differences: count emission for the next timestep (not current one)
Forward-Backward Algorithm

\[
\alpha_1(s) = P(s)P(x_1|s)
\]

\[
\alpha_t(s_t) = \sum_{s_{t-1}} \alpha_{t-1}(s_{t-1})P(s_t|s_{t-1})P(x_t|s_t)
\]

\[
\beta_n(s) = 1
\]

\[
\beta_t(s_t) = \sum_{s_{t+1}} \beta_{t+1}(s_{t+1})P(s_{t+1}|s_t)P(x_{t+1}|s_{t+1})
\]

\[
P(s_{t+1} = 2|x) = \frac{\alpha_3(2)\beta_3(2)}{\sum_i \alpha_3(i)\beta_3(i)}
\]

- Does this explain why beta is what it is?
- What is the denominator here? \(P(x)\)

HMM POS Tagging

- Baseline: assign each word its most frequent tag: ~90% accuracy
- Trigram HMM: ~95% accuracy / 55% on unknown words

Trigram Taggers

- Trigram model: \(y_1 = (<S>, \text{NNP}), y_2 = (\text{NNP}, \text{VBZ}), \ldots\)
- \(P(\text{VBZ}, \text{NN}) | (\text{NNP}, \text{VBZ}))\) — more context! Noun-verb-noun S-V-O
- Tradeoff between model capacity and data size — trigrams are a “sweet spot” for POS tagging

HMM POS Tagging

- Baseline: assign each word its most frequent tag: ~90% accuracy
- Trigram HMM: ~95% accuracy / 55% on unknown words
- TnT tagger (Brants 1998, tuned HMM): 96.2% accuracy / 86.0% on unks
- State-of-the-art (BiLSTM-CRFs): 97.5% / 89%+

Slide credit: Dan Klein
Errors

<table>
<thead>
<tr>
<th>JJ</th>
<th>NN</th>
<th>NNP</th>
<th>NNPS</th>
<th>RB</th>
<th>RP</th>
<th>IN</th>
<th>VB</th>
<th>VBD</th>
<th>VBN</th>
<th>VBP</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>177</td>
<td>56</td>
<td>0</td>
<td>61</td>
<td>2</td>
<td>5</td>
<td>10</td>
<td>15</td>
<td>108</td>
<td>0</td>
<td>488</td>
</tr>
<tr>
<td>244</td>
<td>0</td>
<td>103</td>
<td>0</td>
<td>12</td>
<td>1</td>
<td>1</td>
<td>29</td>
<td>5</td>
<td>6</td>
<td>19</td>
<td>525</td>
</tr>
<tr>
<td>107</td>
<td>106</td>
<td>0</td>
<td>132</td>
<td>5</td>
<td>0</td>
<td>7</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>427</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>110</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>142</td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>21</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>16</td>
<td>138</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>295</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>39</td>
<td>0</td>
<td>65</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>104</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>169</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>323</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>64</td>
<td>9</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>7</td>
<td>85</td>
<td>189</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>143</td>
<td>2</td>
<td>166</td>
</tr>
<tr>
<td>101</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>108</td>
<td>0</td>
<td>1</td>
<td>221</td>
</tr>
<tr>
<td>5</td>
<td>34</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>49</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>104</td>
<td></td>
</tr>
</tbody>
</table>

Total: 626 JJ 536 NN 348 VBP 317 DT 144 NN 122 RB 102 IN 140 VBD 269 VBN 108

J/J/NN NN official knowledge VBD/IN DT NN made up the story RB VBD/VBN NNS recently sold shares (NN NN: tax cut, art gallery, ...)

Remaining Errors

- Lexicon gap (word not seen with that tag in training) 4.5%
- Unknown word: 4.5%
- Could get right: 16% (many of these involve parsing!)
- Difficult linguistics: 20%
  
  VBD / VBP? (past or present?)

  They set up absurd situations, detached from reality

- Underspecified / unclear, gold standard inconsistent / wrong: 58%

  adjective or verbal participle? JJ / VBN?

- Could get right: 16% (many of these involve parsing!)

  Discontinued operations

They set up absurd situations, detached from reality

Manning 2011 “Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics?”

Other Languages

<table>
<thead>
<tr>
<th>Language</th>
<th>CRF+</th>
<th>CRF</th>
<th>BTS</th>
<th>BTS*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bulgarian</td>
<td>97.97</td>
<td>97.00</td>
<td>97.84</td>
<td>97.02</td>
</tr>
<tr>
<td>Czech</td>
<td>98.38</td>
<td>98.00</td>
<td>98.30</td>
<td>98.44</td>
</tr>
<tr>
<td>Danish</td>
<td>95.93</td>
<td>95.06</td>
<td>95.52</td>
<td>92.45</td>
</tr>
<tr>
<td>German</td>
<td>93.08</td>
<td>91.99</td>
<td>92.87</td>
<td>92.34</td>
</tr>
<tr>
<td>Greek</td>
<td>97.72</td>
<td>97.21</td>
<td>97.39</td>
<td>96.64</td>
</tr>
<tr>
<td>English</td>
<td>95.11</td>
<td>94.51</td>
<td>93.87</td>
<td>94.00</td>
</tr>
<tr>
<td>Spanish</td>
<td>96.08</td>
<td>95.03</td>
<td>95.80</td>
<td>95.26</td>
</tr>
<tr>
<td>Farsi</td>
<td>96.59</td>
<td>96.25</td>
<td>96.82</td>
<td>96.76</td>
</tr>
<tr>
<td>Finnish</td>
<td>94.34</td>
<td>92.82</td>
<td>95.48</td>
<td>96.05</td>
</tr>
<tr>
<td>French</td>
<td>96.00</td>
<td>95.93</td>
<td>95.75</td>
<td>95.17</td>
</tr>
<tr>
<td>Indonesian</td>
<td>92.84</td>
<td>92.71</td>
<td>92.85</td>
<td>91.03</td>
</tr>
<tr>
<td>Italian</td>
<td>97.70</td>
<td>97.61</td>
<td>97.56</td>
<td>97.40</td>
</tr>
<tr>
<td>Swedish</td>
<td>96.81</td>
<td>96.15</td>
<td>95.57</td>
<td>93.17</td>
</tr>
</tbody>
</table>

Average: 96.04 BTS 95.41 SPANS 95.85 95.06

- Universal POS tagset (~12 tags), cross-lingual model works as well as tuned CRF using external resources

Next Time

- CRFs: feature-based discriminative models
- Named entity recognition
- CRFs: feature-based discriminative models
- Structured SVM for sequences

Slide credit: Dan Klein / Toutanova + Manning (2000)