Maximum Entropy Models: Your Friends and Mine

Stanley Chen

School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Carnegie Mellon

April 3, 1997

Overview

- What are maximum entropy models, and why they are better than HMM's and decision trees
- Why maximum entropy models will solve all of the world's problems
- Triumphs of maximum entropy
- What's wrong with ME and what I do for a living

Our National Pastime

$$p(\textit{Yankees win}|\textit{home game}) = 0.7$$

$$p(\textit{Yankees win}|\textit{Gooden pitches}) = 0.7$$

$$p(\textit{Yankees win}|\textit{home}, \textit{Gooden}) = ?$$

$$p(\textit{Yankees win}) = 0.7$$

$$p(\textit{Yankees win}|\textit{home}, \textit{Gooden}) = 0.7$$

Our National Pastime

$$p(\mathit{Yankees\ win} | \mathit{home\ game}) = 0.7$$

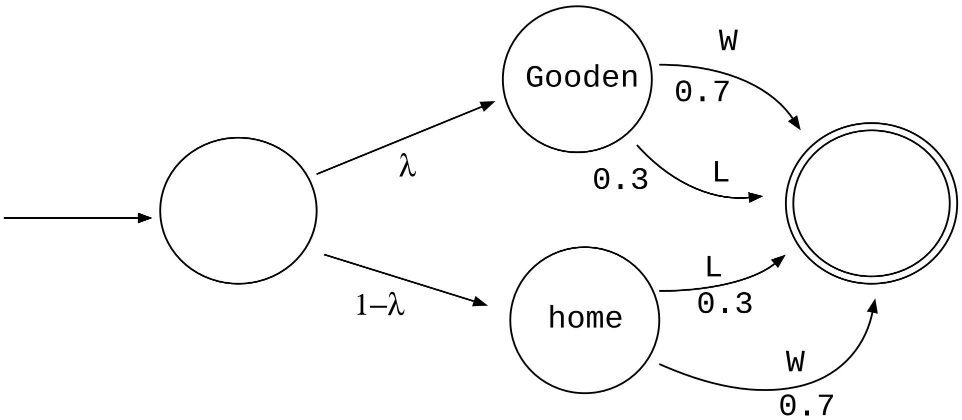
$$p(\mathit{Yankees\ win} | \mathit{Gooden\ pitches}) = 0.7$$

$$p(\mathit{Yankees\ win} | \mathit{home}, \mathit{Gooden}) = \ ?$$

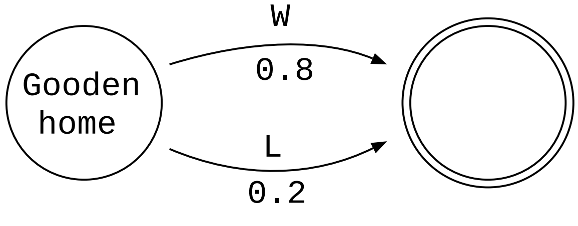
$$p(\mathit{Yankees\ win}) = 0.4$$

$$p(\mathit{Yankees\ win} | \mathit{home}, \mathit{Gooden}) > 0.7$$

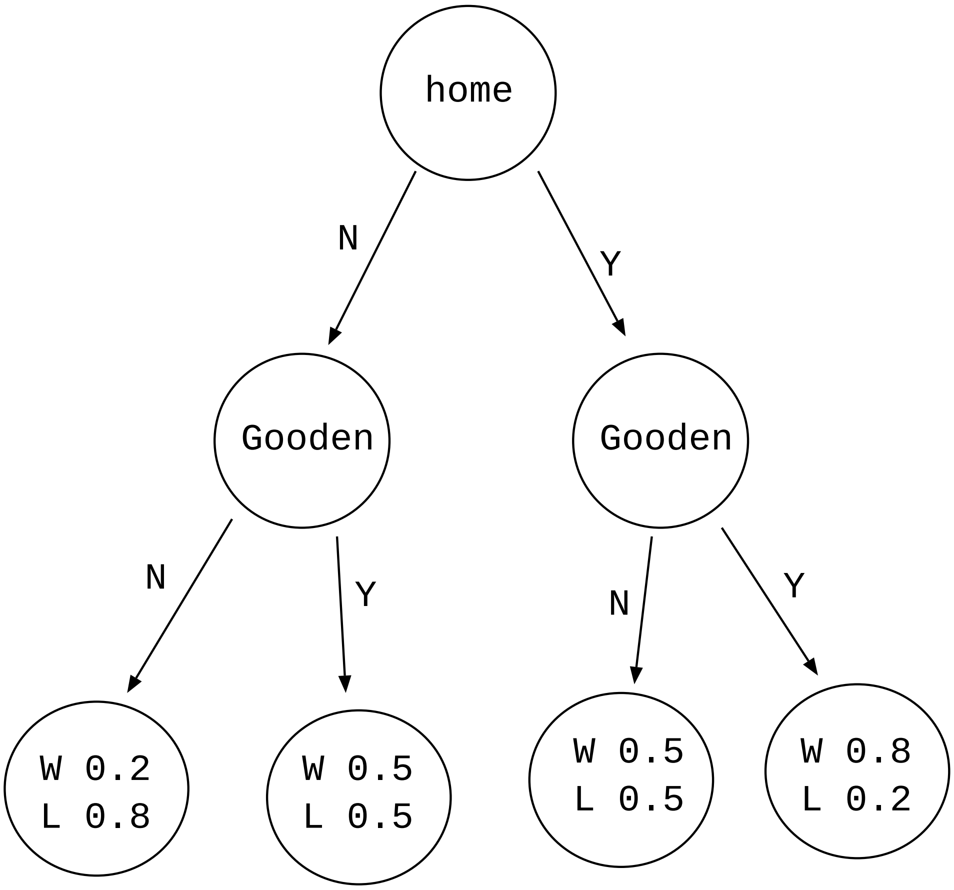
HMM's



$$
\begin{aligned}
& p(\mathit{Yankees\ win}|\mathit{home}, \mathit{Gooden}) = \\
& \quad (1-\lambda) \times p(\mathit{Yankees\ win}|\mathit{home}) + \\
& \quad \lambda \times p(\mathit{Yankees\ win}|\mathit{Gooden}) = \\
& \quad (1-\lambda) \times 0.7 + \lambda \times 0.7 = 0.7
\end{aligned}
$$



Decision Trees



Our National Pastime

problems

- linear interpolation doesn't cut it
- decision-trees can't combine independent information sources efficiently

solution

- exponential models
  - maximum entropy models refer to way of training/selecting an exponential model

Combining Multiple Information Sources

exponential models

- mathematical intuition

$$
\begin{aligned}
p(y|x_1,\ldots,x_n) &= \frac{p(x_1,\ldots,x_n|y)p(y)}{p(x_1,\ldots,x_n)} \\
&\approx p(y)\prod_{i=1}^{n}\frac{p(x_i|y)}{p(x_i)} \\
&= p(y)\prod_{i=1}^{n}\frac{p(y|x_i)}{p(y)}
\end{aligned}
$$

e.g.,

$$
\begin{aligned}
&p(\mathit{win}|\mathit{home},\mathit{Gooden}) \\
&\approx\ p(\mathit{win})\times\frac{p(\mathit{win}|\mathit{home})}{p(\mathit{win})}\times\frac{p(\mathit{win}|\mathit{Gooden})}{p(\mathit{win})} \\
&=\ 0.4\times\frac{0.7}{0.4}\times\frac{0.7}{0.4}=1.225\approx 0.8
\end{aligned}
$$

Exponential Models

$$p(y|x_1,\ldots,x_n) \approx p(y)\prod_{i=1}^{n}\frac{p(y|x_i)}{p(y)}$$

$$\begin{aligned} p(y|\vec{x}) &= \frac{1}{Z(\vec{x})}\prod_{i \text{ active}} e^{\lambda_i} \\ &= \frac{1}{Z(\vec{x})}\exp(\sum_{i \text{ active}} \lambda_i) \end{aligned}$$

$$\begin{aligned} & p(\mathit{win}|\mathit{home}, \mathit{Gooden}) \\ \approx\ & p(\mathit{win}) \times \frac{p(\mathit{win}|\mathit{home})}{p(\mathit{win})} \times \frac{p(\mathit{win}|\mathit{Gooden})}{p(\mathit{win})} \\ =\ & 0.4 \times \frac{0.7}{0.4} \times \frac{0.7}{0.4} = 1.225 \approx 0.8 \end{aligned}$$

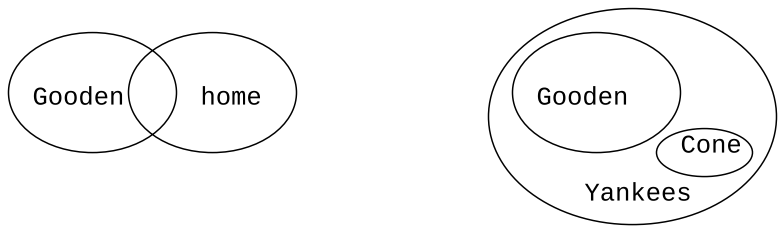
$$\begin{aligned} & p(\mathit{win}|\mathit{home}, \mathit{Gooden}) \\ =\ & \frac{1}{Z}\exp(\lambda_{\text{win}} + \lambda_{\text{home win}} + \lambda_{\text{Gooden win}}) \\ =\ & \frac{1}{Z}\exp(\log 0.4 + \log\frac{0.7}{0.4} + \log\frac{0.7}{0.4}) \\ \approx\ & 0.8 \end{aligned}$$

Maximum Entropy Models

training λ's

- find model $p(y|x)$ satisfying constraints with highest entropy blah blah blah
- find ML λ's
- generalized iterative scaling — guaranteed — convex space

Exponential Models and HMM's and Decision Trees



- if features don't partially overlap, exp models can imitate other models
- if features do partially overlap, exp models can surpass other models

$$\begin{aligned} p(\mathit{win}|\mathit{Yankees}) &= 0.4 \\ p(\mathit{win}|\mathit{Gooden}) &= 0.7 \\ p(\mathit{win}|\mathit{Yankees}, \mathit{Gooden}) &= p(\mathit{win}|\mathit{Gooden}) = 0.7 \end{aligned}$$

$$\begin{aligned} p(\mathit{win}|\mathit{Yankees}) &= \frac{1}{Z_1}\exp(\lambda_{\text{win}} + \lambda_{\text{Y win}}) \\ p(\mathit{win}|\mathit{Gooden}) &= \frac{1}{Z_2}\exp(\lambda_{\text{win}} + \lambda_{\text{Y win}} + \lambda_{\text{G win}}) \\ p(\mathit{win}|\mathit{Yankees}, \mathit{Gooden}) &= \frac{1}{Z_2}\exp(\lambda_{\text{win}} + \lambda_{\text{Y win}} + \lambda_{\text{G win}}) \\ p(\mathit{win}|\mathit{Yankees}, \mathit{Cone}) &= \frac{1}{Z_3}\exp(\lambda_{\text{win}} + \lambda_{\text{Y win}}) \\ &= \frac{Z_1}{Z_3} p(\mathit{win}|\mathit{Yankees}) \end{aligned}$$

Exponential Models and HMM's and Decision Trees

exp models $\supset$ decision trees

- for any decision tree, can build equivalent exp model
- plus get behavior if allow leafs to overlap
- decision-tree technology can be seen as way to produce features for exp models

exp models $\supset$ HMM's

- for any HMM, can build equivalent exp model? (train by alternate EM and GIS)
- plus get behavior if allow leafs to overlap
- HMM technology can be seen as way to produce features for exp models

Exponential Models and the n-Gram Thing

What have you done for me lately?

- trigger models, topic-specific models, class-based models, grammatical models, remote corpus models

combining information sources

- linear interpolation has had limited effectiveness
- some predictions are helped, some are hurt
- with exponential models, less apt to hurt predictions, because information is combined in better way?

Triumphs of ME Models

Roni's work

- n-grams
- n-grams with a skipped word
- triggers
- *i.e.*, things that partially overlap

results

- 32–39% reduction in perplexity over trigram model, 10–14% reduction in WER
- experiments with linear interpolation did much worse
- just about the best LM result ever

Triumphs of ME Models

Finding article boundaries (Beeferman, Berger, Lafferty)

- informedia
- topic models

guess whether article boundary occurs at each sentence boundary

- topic coherence (trigger model)
- whether certain words occur in surrounding sentences, *e.g.*, *incorporated* in next sentence
- (video cues, audio cues)
- *i.e.*, things that partially overlap

Problems with ME and What I Do

infrastructure

- slow evaluation
  - unnormalized models
  - no WER change in n-gram model with triggers
- slow training (200 Alpha-days for 38M words of training)
  - fast training — change nature of constraints
  - n-gram models
- smoothing
  - n-gram models

current work

- more complex models — Roni's work
- feature induction