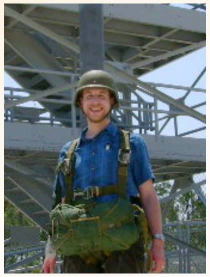
Computationally Efficient M-Estimation of Log-Linear Structure Models



Noah Smith, **Doug Vail**, and **John Lafferty**

School of Computer Science

Carnegie Mellon University

{nasmith,dvail2,lafferty}@cs.cmu.edu

Sketch of the Talk

A new loss function for supervised structured classification with arbitrary features.

- Fast & easy to train - no partition functions!
- Consistent estimator of the joint distribution
- Information-theoretic interpretation
- Some practical issues
- Speed & accuracy comparison

Log-Linear Models as Classifiers

Distribution:

in out

$$p_{\mathbf{w}}(x, y) = \frac{e^{\mathbf{w}^\top \mathbf{f}(x,y)}}{\sum_{x',y'} e^{\mathbf{w}^\top \mathbf{f}(x',y')}} = \frac{e^{\mathbf{w}^\top \mathbf{f}(x,y)}}{Z(\mathbf{w})}$$

parameters

partition function

Classification:

dot-product score

$$class(x) = \arg\max_{y} p_{\mathbf{w}}(x, y) = \arg\max_{y} \mathbf{w}^\top \mathbf{f}(x, y)$$

dynamic programming, search, discrete optimization, etc.

Training Log-Linear Models

Maximum Likelihood Estimation:

$$\begin{aligned}\hat{\mathbf{w}} &= \arg\max_{\mathbf{w}} \sum_{x,y} \tilde{p}(x,y) \log p_{\mathbf{w}}(x,y) \\ &= \arg\max_{\mathbf{w}} \left(\sum_{x,y} \tilde{p}(x,y) \mathbf{w}^\top \mathbf{f}(x,y) \right) - \log Z(\mathbf{w})\end{aligned}$$

pain

Also, *discriminative* alternatives:

- conditional random fields (x-wise partition functions)
- maximum margin training (decoding during training)

Notational Variant

“some other” distribution

$$p_{\mathbf{w}}(x, y) = \frac{q_0(x,y)e^{\mathbf{w}^\top \mathbf{f}(x,y)}}{Z(\mathbf{w},q_0)}$$

Still log-linear. $w_0 = 1; f_0(x, y) = \log q_0(x, y)$

Jeon and Lin (2006)

A new loss function for training:

$$\ell(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{e^{-\mathbf{w}^\top \mathbf{f}(x_i,y_i)}}_{\text{exponentiated, negated dot-product scores}} + \sum_{x,y} \underbrace{q_0(x,y)}_{\text{base distribution}} \left(\mathbf{w}^\top \mathbf{f}(x,y)\right)$$

Jeon and Lin (2006)

A new loss function for training:

$$\ell(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} e^{-\mathbf{w}^\top \mathbf{f}(x_i, y_i)} + \sum_{x,y} q_0(x,y)\left(\mathbf{w}^\top \mathbf{f}(x,y)\right)$$

exponentiated, negated dot-product scores

base distribution

$$\ell(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} e^{-\mathbf{w}^\top \mathbf{f}(x_i, y_i)} + \mathbf{w}^\top \underbrace{\left(\sum_{x,y} q_0(x,y)\mathbf{f}(x,y)\right)}_{\mathbf{E}_{q_0}[\mathbf{f}]}$$

Attractive Properties of the M-Estimator

✓Computationally efficient.

$$O\left(\sum_{i=1}^{n}\sum_{j=1}^{m}[f_j(x_i, y_i) \neq 0]\right)$$

$$O\left(m\right)$$

$$\ell(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} e^{-\mathbf{w}^\top \mathbf{f}(x_i, y_i)} + \mathbf{w}^\top \underbrace{\left(\sum_{x,y} q_0(x, y)\mathbf{f}(x, y)\right)}_{\mathbf{E}_{q_0}[\mathbf{f}]}$$

Attractive Properties of the M-Estimator

✓ Convex.

$\exp$ is convex; affine composition $\longrightarrow$ convex

linear

linear

$$\ell(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} e^{-\mathbf{w}^\top \mathbf{f}(x_i,y_i)} + \mathbf{w}^\top \underbrace{\left(\sum_{x,y} q_0(x,y)\mathbf{f}(x,y)\right)}_{\mathbf{E}_{q_0}[\mathbf{f}]}$$

Statistical Consistency

- If the data were drawn from some distribution in the given family, parameterized by $\mathbf{w}^*$, then

$$\forall \epsilon > 0, \lim_{n \to \infty} \Pr\left(\left|\left(\arg\min_{\mathbf{w}} \ell(\mathbf{w})\right) - \mathbf{w}^*\right| < \epsilon\right) = 1$$

- True of MLE, Pseudolikelihood, and the M-estimator.
  - Conditional likelihood is consistent for the conditional distribution.

Information-Theoretic Interpretation

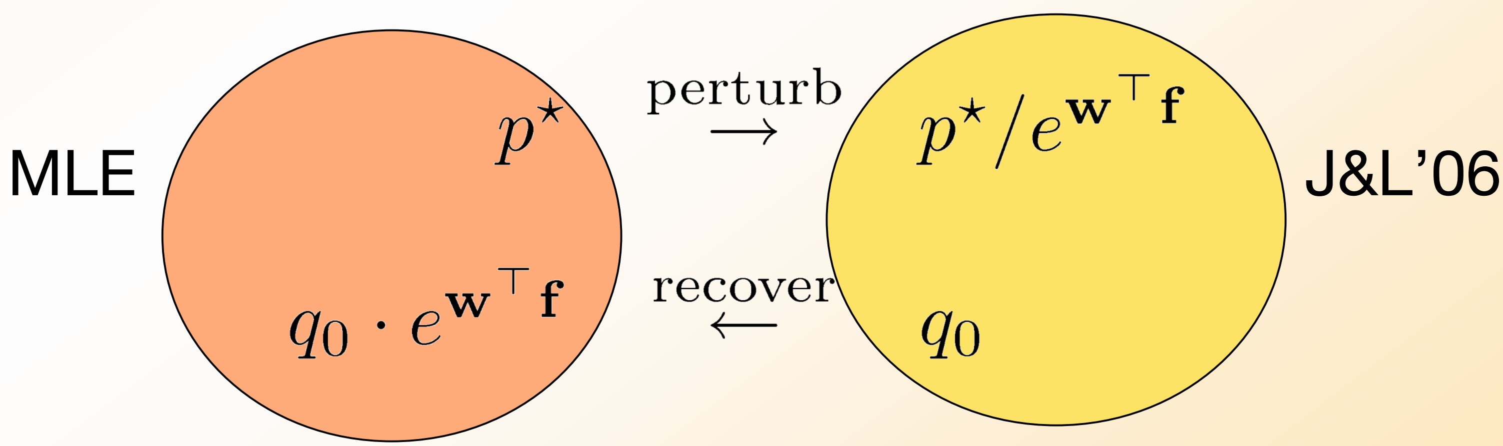
- True model: $p^\star$
- Perturbation applied to $p^\star$, resulting in q_0
- Goal: *recover* the true distribution by correcting the perturbation.

$$p^\star \quad \overset{\text{perturb}}{\longrightarrow} \quad p^\star / e^{\mathbf{w}^\top \mathbf{f}}$$

$$q_0 \cdot e^{\mathbf{w}^\top \mathbf{f}} \quad \overset{\text{recover}}{\longleftarrow} \quad q_0$$

Information-Theoretic Interpretation

- True model: $p^\star$
- Perturbation applied to $p^\star$, resulting in q_0
- Goal: *recover* the true distribution by correcting the perturbation.



Minimizing KL Divergence

$$
\begin{aligned}
& D(q_0 \| p^\star \cdot e^{-\mathbf{w}^\top \mathbf{f}}) \\
&= \sum_{x,y} q_0(x,y) \log \frac{q_0(x,y)}{p^\star(x,y) e^{-\mathbf{w}^\top \mathbf{f}(x,y)}} + p^\star(x,y) e^{-\mathbf{w}^\top \mathbf{f}(x,y)} - q_0(x,y) \\
&= \sum_{x,y} p^\star(x,y) e^{-\mathbf{w}^\top \mathbf{f}(x,y)} + q_0(x,y) \mathbf{w}^\top \mathbf{f}(x,y) + \text{constant}(\mathbf{w}) \\
&\approx \frac{1}{n} \sum_{i=1}^{n} e^{-\mathbf{w}^\top \mathbf{f}(x_i, y_i)} + \mathbf{w}^\top \mathbf{E}_{q_0}[\mathbf{f}] + \text{constant}(\mathbf{w}) \\
&= \ell(\mathbf{w}) + \text{constant}(\mathbf{w})
\end{aligned}
$$

$$p^\star \xrightarrow{\text{perturb}} p^\star / e^{\mathbf{w}^\top \mathbf{f}}$$

$$q_0 \cdot e^{\mathbf{w}^\top \mathbf{f}} \xleftarrow{\text{recover}} q_0$$

So far …

- Alternative objective function for log-linear models.
  - Efficient to compute
  - Convex and differentiable
  - Easy to implement
  - Consistent
- Interesting information-theoretic motivation.

Next …

- Practical issues
- Experiments

q_o Desiderata

- Fast to estimate
- Smooth
- Straightforward calculation of $\mathbf{E}_{q^o}[\mathbf{f}]$

$$p_{\mathbf{w}}(x, y) = \frac{q_0(x,y)e^{\mathbf{w}^\top \mathbf{f}(x,y)}}{Z(\mathbf{w},q_0)}$$

Here: smoothed HMM.

- See paper for details on $\mathbf{E}_{q^o}[\mathbf{f}]$ - linear system!

In general, can *sample* from q_o to estimate.

Optimization

Can use Quasi-Newton methods (L-BFGS, CG).

The gradient:

$$\frac{\partial \ell}{\partial w_j} = -\frac{1}{n}\sum_{i=1}^{n} e^{-\mathbf{w}^\top \mathbf{f}(x_i, y_i)} f_j(x_i, y_i) + \mathbf{E}_{q_0}[f_j]$$

Regularization

Problem: If we estimate $\mathbf{E}_{q^\circ}[f_j] = 0$, then w_j will tend toward -∞.

Quadratic regularizer: $$\min_{\mathbf{w}} \ell(\mathbf{w}) + \frac{\mathbf{w}^\top \mathbf{w}}{2c}$$

Can be interpreted as a 0-mean, c-variance, diagonal Gaussian prior on $\mathbf{w}$; maximum *a posteriori* analog for the M-estimator.

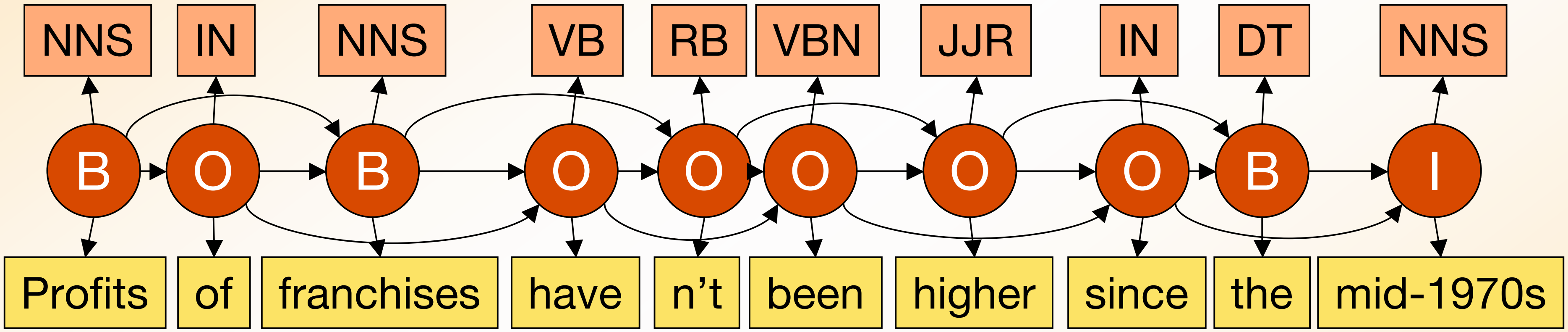
Experiments

- Data: CoNLL-2000 shallow parsing dataset
- Task: NP-chunking (by B-I-O labeling)

- Baseline/q_0: smoothed MLE trigram HMM; B-I-O label emits word and tag separately

- Quadratic regularization for log-linear models, c selected on held-out.

B-I-O Example



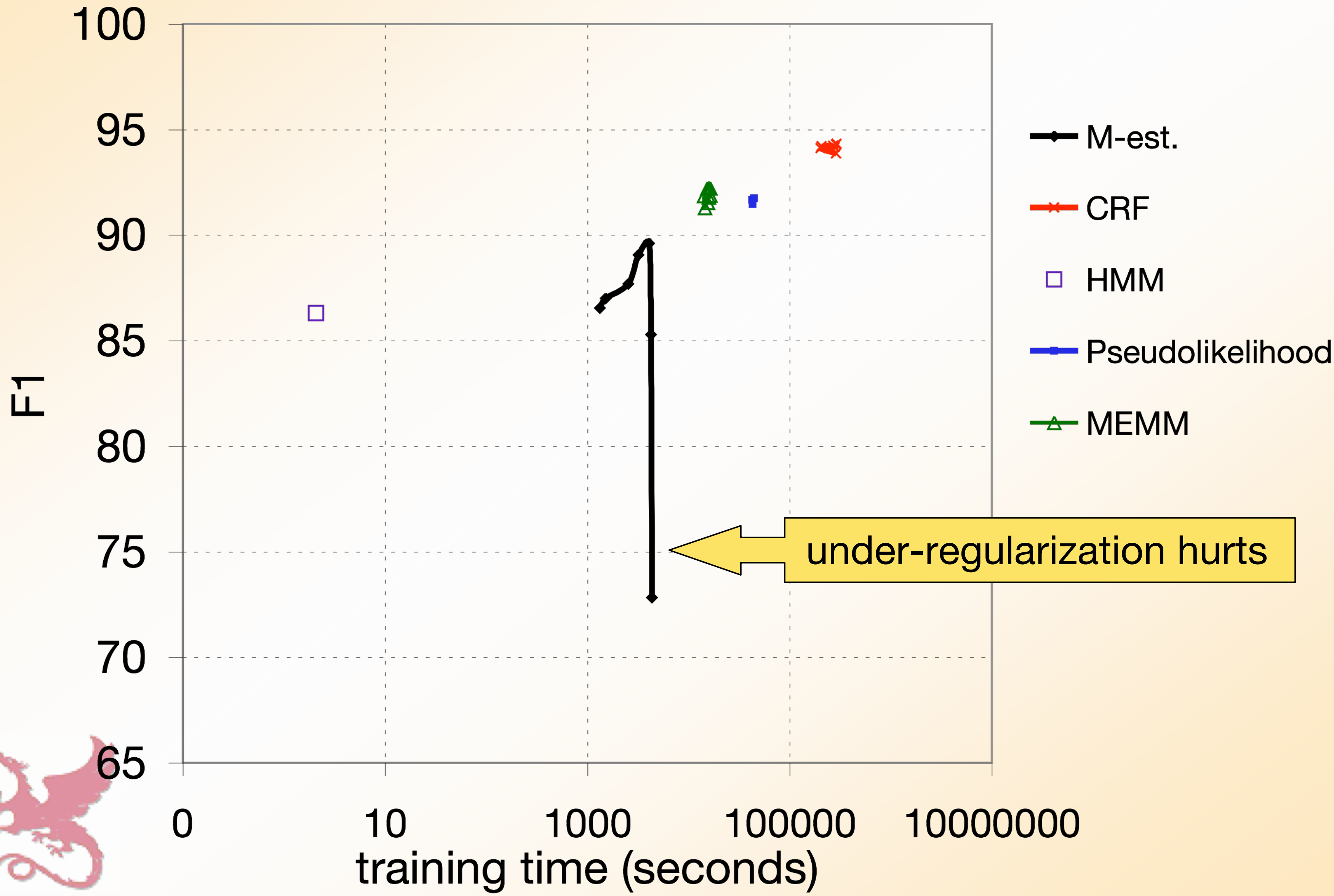
NNS
IN
NNS
VB
RB
VBN
JJR
IN
DT
NNS
B
O
B
O
O
O
O
O
B
I
Profits
of
franchises
have
n't
been
higher
since
the
mid-1970s

Experiments

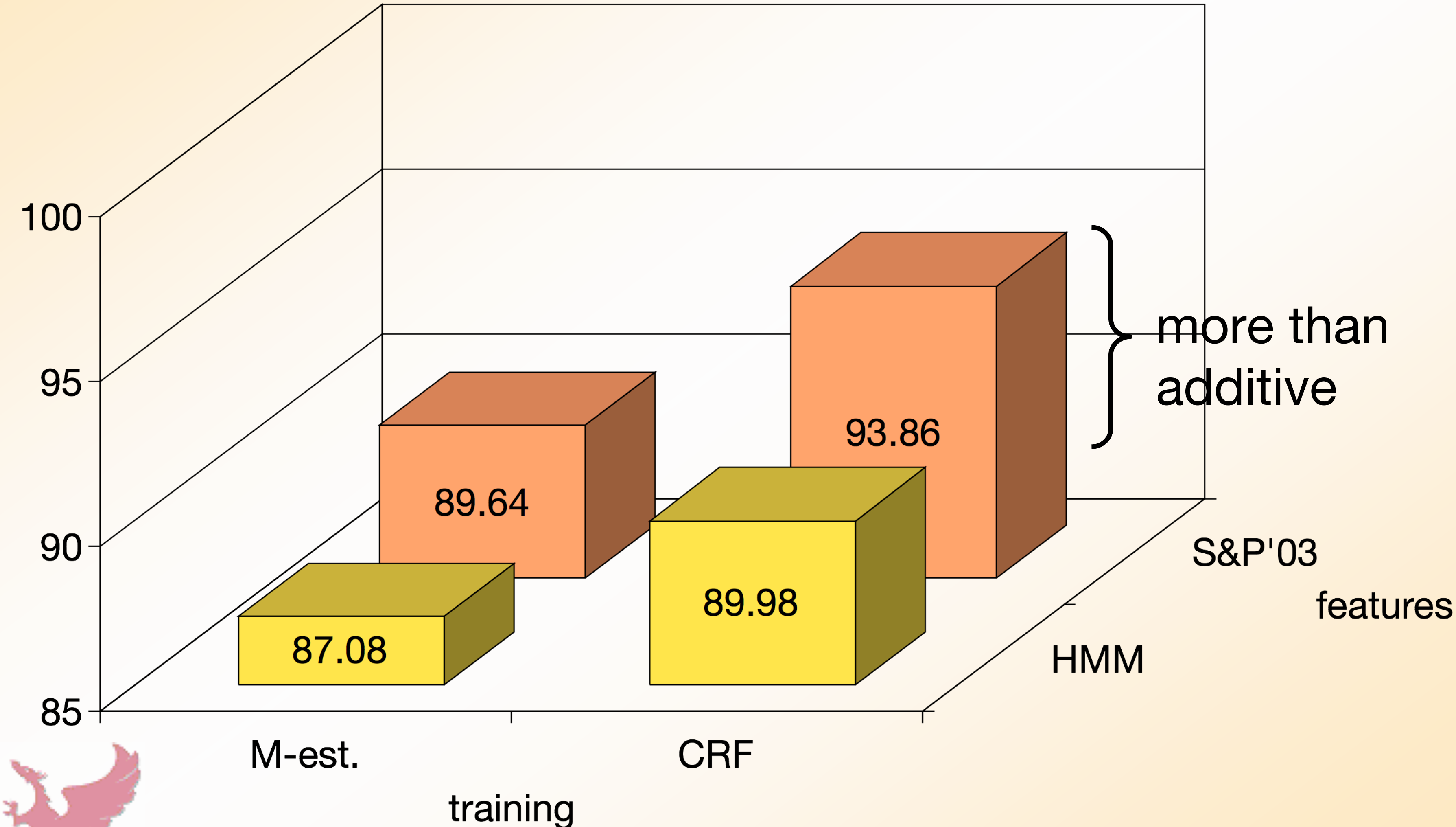
	time (h:m:s)	precision	recall	F_1
HMM	0:00:02	85.6	88.7	87.1
M-est.	1:01:37	88.9	90.4	89.6
MEMM	3:39:52	90.9	92.2	91.5
PL	9:34:52	91.9	91.8	91.8
CRF	64:18:24	94.0	93.7	93.9

rich features
(Sha & Pereira '03)

Accuracy, Training Time, and c



Generative/Discriminative vs. Features



18 Minutes Are Not Enough

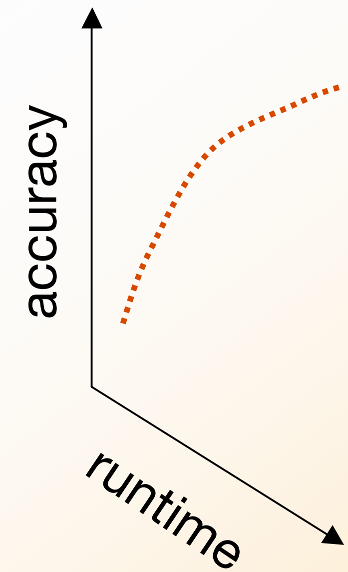
- See the paper
  - q_o experiments
  - negative result: attempt to "make it discriminative"
- WSJ section 22 dependency parsing
  - generative baseline/q_o ($\approx$ Klein & Manning '03)
  - 85.2% $\rightarrow$ 86.4%
  - 2 million $\rightarrow$ 3 million features ($\approx$ McDonald et al. '05)
  - 4 hours training per value of c

Ongoing & Future Work

- **Discriminative training** works better but takes longer.
  - Cases where discriminative training may be too expensive
    - high complexity inference (parsing)
    - n is very large (MT?)
  - Is there an efficient estimator like this for the *conditional* distribution?
- **Hidden variables** increase complexity, too.
  - Use M-estimator for M step in EM?
  - Is there an efficient estimator like this that handles hidden variables?

Conclusion

- M-estimation is
  - fast to train (no partition functions)
  - easy to implement
  - statistically consistent
  - feature-empowered (like CRFs)
  - generative



A new point on the spectrum of speed/accuracy/expressiveness tradeoffs.

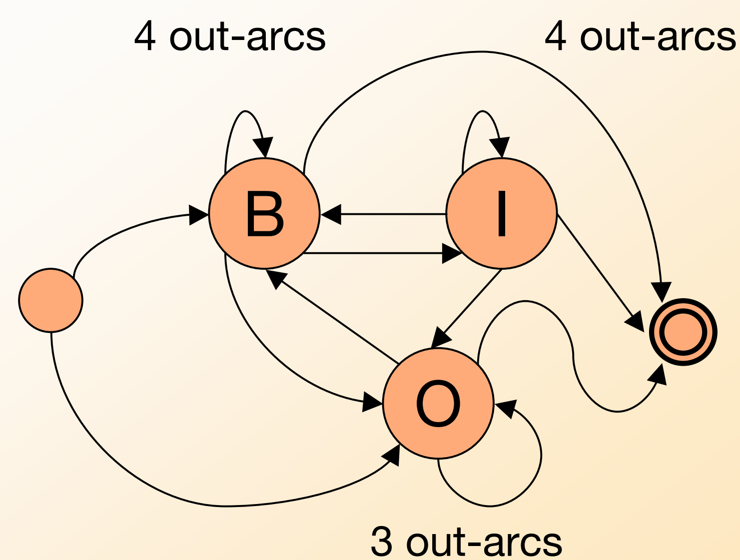
Thanks!

How important is the choice of q_0?

- MAP-trained HMM
- Empirical marginal:

$$q_0'(\text{words}, \text{tags}, \text{labels}) = q_0(\text{labels} \mid \text{words}, \text{tags}) \cdot \tilde{p}(\text{words}, \text{tags})$$

- Locally uniform model
  - Uniform transitions
  - No temporal effects
  - 0% precision, recall



q_o Experiments

q_o	select c to maximize:	precision	recall	F_1
baseline HMM (no M-est.)		85.6	88.7	87.1
HMM	F_1	**88.9**	**90.4**	**89.6**
empirical marginal	F_1	84.4	**89.4**	86.8
locally uniform transitions	F_1	72.9	57.6	64.3
	precision	84.4	37.7	52.1

Negative Result: Input-Only Features

Idea: Make M-estimator "more discriminative" by including features of words/tags only.

- Think of the model in two parts:

$$p_{\mathbf{w}}(\text{words}, \text{tags}, \text{labels}) = p_{\mathbf{w}_1^k}(\text{labels} \mid \text{words}, \text{tags}) \cdot p_{\mathbf{w}_{k+1}^m}(\text{words}, \text{tags})$$

Improve fit here …

… by doing more of the "explanatory work" here.

→ Virtually no effect.