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
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:

\[ p_w(x, y) = \frac{e^{w^\top f(x, y)}}{\sum_{x', y'} e^{w^\top f(x', y')}} = \frac{e^{w^\top f(x, y)}}{Z(w)} \]

Classification:

\[ \text{class}(x) = \arg \max_y p_w(x, y) = \arg \max_y w^\top f(x, y) \]

dynamic programming, search, discrete optimization, etc.
Training Log-Linear Models

Maximum Likelihood Estimation:

\[
\hat{\mathbf{w}} = \arg \max_{\mathbf{w}} \sum_{x,y} \tilde{p}(x, y) \log p_\mathbf{w}(x, y)
\]

\[
= \arg \max_{\mathbf{w}} \left( \sum_{x,y} \tilde{p}(x, y) \mathbf{w}^\top \mathbf{f}(x, y) \right) - \log Z(\mathbf{w})
\]

Also, discriminative alternatives:

• conditional random fields \((x\text{-wise partition functions})\)
• maximum margin training \((\text{decoding during training})\)
Notational Variant

\[ p_w(x, y) = \frac{q_0(x, y)e^{w^\top f(x, y)}}{Z(w, q_0)} \]

"some other" distribution

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:

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

- exponentiated, negated dot-product scores
- base distribution
Jeon and Lin (2006)

A new loss function for training:

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

\[ \ell(w) = \frac{1}{n} \sum_{i=1}^{n} e^{-w^\top f(x_i, y_i)} + \sum_{x, y} q_0(x, y) f(x, y) - \mathbb{E}_{q_0}[f] \]
Attractive Properties of the M-Estimator

✓ Computationally efficient.

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

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

\[
O(m)
\]

\[
E_{q_0}[f]
\]
Attractive Properties of the M-Estimator

✓ Convex.

\[
\ell(w) = \frac{1}{n} \sum_{i=1}^{n} e^{-w^T f(x_i, y_i)} + w^T \left( \sum_{x, y} q_0(x, y) f(x, y) \right) - E_{q_0}[f]
\]

- \(\exp\) is convex; affine composition \(\rightarrow\) convex

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

- linear
Statistical Consistency

• If the data were drawn from some distribution in the given family, parameterized by $w^*$, then

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

• True of MLE, Pseudolikelihood, and the M-estimator.
  – Conditional likelihood is consistent for the conditional distribution.
Information-Theoretic Interpretation

• True model: \( p^* \)
• Perturbation applied to \( p^* \), resulting in \( q_0 \)
• Goal: recover the true distribution by correcting the perturbation.

\[
p^* \xrightarrow{\text{perturb}} p^*/e^{w^T f} \]

\[
q_0 \cdot e^{w^T f} \xleftarrow{\text{recover}} q_0
\]
Information-Theoretic Interpretation

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

\[
D(q_0\|p^* \cdot e^{-\mathbf{w}^\top \mathbf{f}}) = \sum_{x,y} q_0(x,y) \log \frac{q_0(x,y)}{p^*(x,y)e^{-\mathbf{w}^\top \mathbf{f}(x,y)}} + p^*(x,y)e^{-\mathbf{w}^\top \mathbf{f}(x,y)} - q_0(x,y)
\]

\[
= \sum_{x,y} p^*(x,y)e^{-\mathbf{w}^\top \mathbf{f}(x,y)} + q_0(x,y)\mathbf{w}^\top \mathbf{f}(x,y) + \text{constant}(\mathbf{w})
\]

\[
\approx \frac{1}{n} \sum_{i=1}^{n} e^{-\mathbf{w}^\top \mathbf{f}(x_i,y_i)} + \mathbf{w}^\top \mathbf{E}_{q_0}[\mathbf{f}] + \text{constant}(\mathbf{w})
\]

\[
= \ell(\mathbf{w}) + \text{constant}(\mathbf{w})
\]

\[\begin{align*}
p^* & \quad \xrightarrow{\text{perturb}} \quad p^*/e^\mathbf{w}^\top \mathbf{f} \\
q_0 \cdot e^{\mathbf{w}^\top \mathbf{f}} & \quad \xrightarrow{\text{recover}} \quad q_0
\end{align*}\]
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
$q_0$ Desiderata

- Fast to estimate
- Smooth
- Straightforward calculation of $E_{q_0}[f]$

Here: smoothed HMM.

- See paper for details on $E_{q_0}[f]$ - linear system!

In general, can sample from $q_0$ to estimate.
Optimization

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

The gradient:

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

**Problem:** If we estimate $\mathbb{E}_{q_0}[f_j] = 0$, then $w_j$ will tend toward $-\infty$.

**Quadratic regularizer:**

$$\min_w \ell(w) + \frac{w^\top w}{2c}$$

Can be interpreted as a 0-mean, $c$-variance, diagonal Gaussian prior on $w$; 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

Profits of franchises have n’t been higher since the mid-1970s
# Experiments

<table>
<thead>
<tr>
<th>Model</th>
<th>Time (h:m:s)</th>
<th>Precision</th>
<th>Recall</th>
<th>$F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM</td>
<td>0:00:02</td>
<td>85.6</td>
<td>88.7</td>
<td>87.1</td>
</tr>
<tr>
<td>M-est.</td>
<td>1:01:37</td>
<td>88.9</td>
<td>90.4</td>
<td>89.6</td>
</tr>
<tr>
<td>MEMM</td>
<td>3:39:52</td>
<td>90.9</td>
<td>92.2</td>
<td>91.5</td>
</tr>
<tr>
<td>PL</td>
<td>9:34:52</td>
<td>91.9</td>
<td>91.8</td>
<td>91.8</td>
</tr>
<tr>
<td>CRF</td>
<td>64:18:24</td>
<td>94.0</td>
<td>93.7</td>
<td>93.9</td>
</tr>
</tbody>
</table>

The table above summarizes the performance of different models in terms of precision, recall, and $F_1$ score. The models include HMM, M-est., MEMM, PL, and CRF. The time each model took to complete the task is also provided. Rich features (Sha & Pereira ‘03) are used in some of the models.
Accuracy, Training Time, and $c$

under-regularization hurts
Generative/Discriminative vs. Features

more than additive

M-est. 87.08 89.64 CRF 89.98 93.86

HMM

S&P'03 features
18 Minutes Are Not Enough

- See the paper
  - $q_0$ experiments
  - negative result: attempt to “make it discriminative”

- WSJ section 22 dependency parsing
  - generative baseline/$q_0$ ($\approx$ 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:
  \[ q'_0(\text{words, tags, labels}) = q_0(\text{labels} \mid \text{words, tags}) \cdot \tilde{p}(\text{words, tags}) \]
- Locally uniform model
  - Uniform transitions
  - No temporal effects
  - 0% precision, recall

[Diagram showing HMM states and transitions]

4 out-arcs
3 out-arcs
### Experiments

<table>
<thead>
<tr>
<th>$q_0$</th>
<th>select $c$ to maximize:</th>
<th>precision</th>
<th>recall</th>
<th>$F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline HMM (no M-est.)</td>
<td></td>
<td>85.6</td>
<td>88.7</td>
<td>87.1</td>
</tr>
<tr>
<td>HMM</td>
<td>$F_1$</td>
<td>88.9</td>
<td>90.4</td>
<td>89.6</td>
</tr>
<tr>
<td>empirical marginal</td>
<td>$F_1$</td>
<td>84.4</td>
<td>89.4</td>
<td>86.8</td>
</tr>
<tr>
<td>locally uniform transitions</td>
<td>$F_1$</td>
<td>72.9</td>
<td>57.6</td>
<td>64.3</td>
</tr>
<tr>
<td></td>
<td>precision</td>
<td>84.4</td>
<td>37.7</td>
<td>52.1</td>
</tr>
</tbody>
</table>
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:

\[ p_w(\text{words, tags, labels}) = p_{w_1}^k(\text{labels | words, tags}) \cdot p_{w_{k+1}}^m(\text{words, tags}) \]

→ Virtually no effect.

... by doing more of the “explanatory work” here.