Natural Language Processing

Language Modeling III
Dan Klein – UC Berkeley
Improving on N-Grams?

- N-grams don’t combine multiple sources of evidence well

\[ P(\text{construction} \mid \text{After the demolition was completed, the}) \]

- Here:
  - “the” gives syntactic constraint
  - “demolition” gives semantic constraint
  - Unlikely the interaction between these two has been densely observed in this specific n-gram

- We’d like a model that can be more statistically efficient
Maximum Entropy Models
Some Definitions

INPUTS

X_i

close the ___

CANDIDATE SET

y(x)

{door, table, ...}

CANDIDATES

y

table

TRUE OUTPUTS

y_i*

door

FEATURE VECTORS

f(x, y) = [0 0 1 0 0 0 1 0 0 0 0 0]

x_1 = “the” ∧ y = “door”

“close” in x ∧ y = “door”

x_1 = “the” ∧ y = “table”

y occurs in x
More Features, Less Interaction

\[ x = \text{closing the } \_\_\_, \ y = \text{doors} \]

- **N-Grams** \[ x_{-1} = \text{“the”} \land y = \text{“doors”} \]
- **Skips** \[ x_{-2} = \text{“closing”} \land y = \text{“doors”} \]
- **Lemmas** \[ x_{-2} = \text{“close”} \land y = \text{“door”} \]
- **Caching** \[ y \text{ occurs in } x \]
## Data: Feature Impact

<table>
<thead>
<tr>
<th>Features</th>
<th>Train Perplexity</th>
<th>Test Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 gram indicators</td>
<td>241</td>
<td>350</td>
</tr>
<tr>
<td>1-3 grams</td>
<td>126</td>
<td>172</td>
</tr>
<tr>
<td>1-3 grams + skips</td>
<td>101</td>
<td>164</td>
</tr>
</tbody>
</table>
Exponential Form

- Weights \( w \)  
- Features \( f(x, y) \)

- Linear score \( w^\top f(x, y) \)

- Unnormalized probability

\[
P(y|x, w) \propto \exp(w^\top f(x, y))
\]

- Probability

\[
P(y|x, w) = \frac{\exp(w^\top f(x, y))}{\sum_{y'} \exp(w^\top f(x, y'))}
\]
Likelihood Objective

- Model form:

\[ P(y|x, w) = \frac{\exp(w^T f(y))}{\sum_{y'} \exp(w^T f(y'))} \]

- Likelihood of training data

\[ L(w) = \log \prod_i P(y_i^*|x_i, w) = \sum_i \log \left( \frac{\exp(w^T f_i(y_i^*))}{\sum_y \exp(w^T f_i(y))} \right) \]

\[ = \sum_i \left( w^T f_i(y_i^*) - \log \sum_y \exp(w^T f_i(y)) \right) \]
Training
History of Training

- **1990’s:** Specialized methods (e.g. iterative scaling)

- **2000’s:** General-purpose methods (e.g. conjugate gradient)

- **2010’s:** Online methods (e.g. stochastic gradient)
What Does LL Look Like?

- **Example**
  - Data: xxxy
  - Two outcomes, x and y
  - One indicator for each
  - Likelihood

\[
\log \left( \left( \frac{e^x}{e^x + e^y} \right)^3 \times \frac{e^y}{e^x + e^y} \right)
\]
The maxent objective is an unconstrained convex problem

\[ L(w) \]

- One optimal value*, gradients point the way
Gradients

\[ L(w) = \sum_i \left( w^T f(x_i, y_i^*) - \log \sum_y \exp(w^T f(x_i, y)) \right) \]

\[ \frac{\partial L(w)}{\partial w} = \sum_i \left( f(x_i, y_i^*) - \sum_y P(y|x_i) f(x_i, y) \right) \]

Count of features under target labels

Expected count of features under model predicted label distribution
Gradient Ascent

- The maxent objective is an unconstrained optimization problem

\[ L(w) \]

- Basic idea: move uphill from current guess
- Gradient ascent / descent follows the gradient incrementally
- At local optimum, derivative vector is zero
- Will converge if step sizes are small enough, but not efficient
- All we need is to be able to evaluate the function and its derivative
(Quasi)-Newton Methods

- 2\textsuperscript{nd}-Order methods: repeatedly create a quadratic approximation and solve it

\[ L(w) \]

\[ L(w_0) + \nabla L(w)^\top(w - w_0) + (w - w_0)^\top \nabla^2 L(w)(w - w_0) \]

- E.g. LBFGS, which tracks derivative to approximate (inverse) Hessian
Regularization
Regularization Methods

- Early stopping

- $L2: LL(w) - |w|^2$

- $L1: LL(w) - |w|$
Regularization Effects

- Early stopping: don’t do this

- L2: weights stay small but non-zero

- L1: many weights driven to zero
  - Good for sparsity
  - Usually bad for accuracy for NLP
Scaling
Why is Scaling Hard?

\[ L(w) = \sum_{i} \left( w^T f(x_i, y_i^*) - \log \sum_y \exp(w^T f(x_i, y)) \right) \]

- Big normalization terms
- Lots of data points
Hierarchical Prediction

- Hierarchical prediction / softmax [Mikolov et al 2013]

- Noise-Contrastive Estimation [Mnih, 2013]

- Self-Normalization [Devlin, 2014]
Stochastic Gradient

- View the gradient as an average over data points

\[
\frac{\partial L(w)}{\partial w} = \frac{1}{N} \sum_i \left( f(x_i, y_i^*) - \sum_y P(y|x_i) f(x_i, y) \right)
\]

- Stochastic gradient: take a step each example (or mini-batch)

\[
\frac{\partial L(w)}{\partial w} \approx \frac{1}{1} \left( f(x_i, y_i^*) - \sum_y P(y|x_i) f(x_i, y) \right)
\]

- Substantial improvements exist, e.g. AdaGrad (Duchi, 11)
Other Methods
Neural Net LMs

\[ i\text{-th output} = P(w_t = i \mid \text{context}) \]

Image: (Bengio et al, 03)
Neural vs Maxent

- Maxent LM

\[ P(y|x, w) \propto \exp(w^\top f(x, y)) \]

- Neural Net LM

\[ P(y|x, w) \propto \exp \left( B\sigma \left( A f(x) \right) \right) \]

\( \sigma \) nonlinear, e.g. tanh
Mixed Interpolation

- But can’t we just interpolate:
  - $P(w|\text{most recent words})$
  - $P(w|\text{skip contexts})$
  - $P(w|\text{caching})$
  - ...

- Yes, and people do (well, did)
  - But additive combination tends to flatten distributions, not zero out candidates
Decision Trees / Forests

- Decision trees?
  - Good for non-linear decision problems
  - Random forests can improve further [Xu and Jelinek, 2004]
  - Paths to leaves basically learn conjunctions
  - General contrast between DTs and linear models