Large Margin Estimation of n-gram Language Models for Speech Recognition via Linear Programming

Vladimir Magdin, Hui Jiang (Department of Computer Science and Engineering, York University, Canada)

Abstract

- Contribution: a novel discriminative training algorithm for backoff *n*-gram language models for use in LVCSR
- The LME-based objective function uses a metric between correct transcriptions and word-graph-encoded competing hypotheses
- The nonlinear LME objective function is approximated by a linear function of LM parameters, which leads to a linear programming solution
- Experimental results on the SPINE1 speech recognition task show a relative reduction in word error rate of close to 2.5%

Language Models in ASR

Recognition is performed via the MAP decision rule:

$$\hat{W} = \arg\max_{W} \Pr(W|X) = \arg\max_{W} \Pr(X|W) \cdot \Pr(W|X)$$

W - sequence of word labels

X - sequence of acoustic observations

Language models in automatic speech recognition:

- LMs constrain the search space of hypotheses
- Pr(W) is modeled via *n*-gram LMs (e.g. Katz back-off LM)

Language model issues:

- ML criterion not directly related to recognition performance
- *n*-gram LMs not tailored for a particular application
- *n*-gram LM parameters are crudely approximated via smoothing

Discriminative Training of LMs via Soft-Margin LME

Based on the principle of large margin estimation, maximize the minimum margin between correct transcription and competing hypotheses:

$$d(W|\Lambda) = \ln\left[\Pr(W|\Lambda) \cdot \mathcal{A}(W)\right] - \max_{W' \in \mathcal{G} \setminus W} \ln\left[\Pr(W'|\Lambda)\right]$$

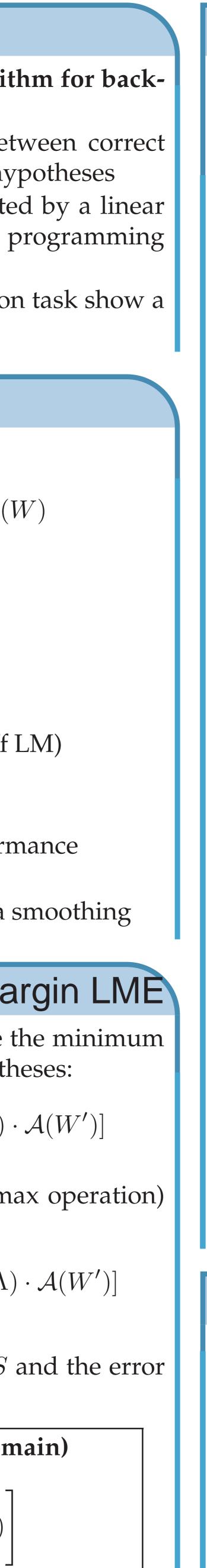
Replacing the maximization with a log-summation (soft-max operation) for mathematical reasons:

$$d(W|\Lambda) = \ln\left[\Pr(W|\Lambda) \cdot \mathcal{A}(W)\right] - \ln\sum_{W' \in \mathcal{G} \setminus W} \ln\left[\Pr(W'|\Lambda)\right]$$

Incorporating the minimum margin over the support set *S* and the error over the error set \mathcal{E} :

Original Soft-Margin LME Objective Function (log-domain)

 $\arg \max_{\Lambda} \left| \min_{W_n \in S} d(W_n | \Lambda) - \epsilon \cdot \frac{1}{|\mathcal{E}|} \sum_{W_i \in \mathcal{E}} d(W_i | \Lambda) \right|$



Solving Soft-Margin LME via Linear Programming

$$\Lambda = \{\lambda_i, \eta_j, \mu_k, \phi_l, \psi_m | i \in P_3, j \in P_2, k \in P_1, l \in Q_2, m \in Q_1\}$$

APPROXIMATION STEP: Approximate original objective function with a simpler linear function

$$\begin{array}{l} \textbf{CORRECT TRANSCRIPTION:}\\ n[\Pr(W|\Lambda) \cdot \mathcal{A}(W)] = \ln \prod_{l=1}^{R_{W}} \Pr(w_{l}|w_{l-2}w_{l-1}) + B'\\ &= \sum_{t=1}^{R_{W}} \ln \Pr(w_{l}|w_{l-2}w_{l-1}) + B'\\ &= \sum_{t=1}^{R_{W}} \ln \Pr(w_{t}|w_{l-2}w_{l-1}) + B'\\ &= \sum_{i \in P_{3}} a'_{i}(W) \cdot \lambda_{i} + \dots + \sum_{m \in Q_{1}} e'_{m}(W) \cdot \psi_{m} + B'\\ \bullet B' \text{ is a constant related to acoustic scores}\\ \bullet a'_{i} \text{ through } e'_{m} \text{ denote counts of individual LM parameters} \end{array}$$

OPTIMIZATION STEP: Maximize approximate objective function to obtain an improved LM

$$\begin{split} \overbrace{d(W|\Lambda) = \sum_{i \in P_3} a_i(W, \mathcal{G}) \cdot \lambda_i + \dots + \sum_{m \in Q_1} e_m(W, \mathcal{G}) \cdot \psi_m \\ a_i(W, \mathcal{G}) = a'_i(W) - a''_i(\mathcal{G}), \text{ etc...}} \\ \mathbf{arg max}_{\Lambda} \left[\underbrace{\min_{W_n \in S} \tilde{d}(W_n | \Lambda) - \epsilon \cdot \frac{1}{|\mathcal{E}|} \sum_{W_i \in \mathcal{E}} \tilde{d}(W_i | \Lambda)}_{W_i \in \mathcal{E}} \tilde{d}(W_i | \Lambda) \right]} \\ \end{split}$$

Experiments

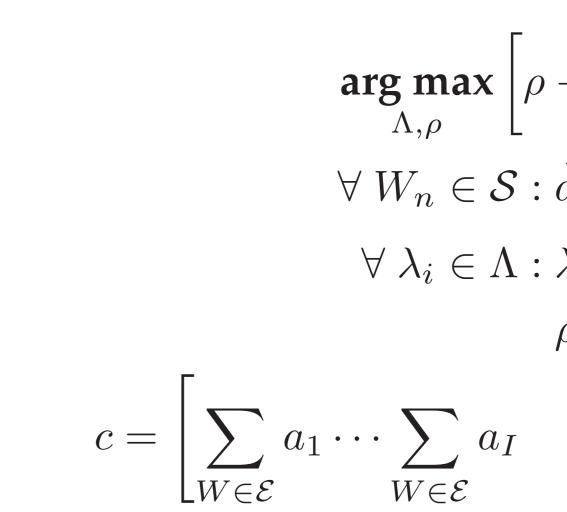
- The discriminative training algorithm was evaluated using the (SPINE1) data set (Quiet subset): 5210 training, 2030 test utterances
- The LM was built using the CMU-Cambridge Statistical Language Modeling toolkit: 1210 unigrams, 12880 bigrams, and 27924 trigrams
- HMM model training, wordgraph generation, and recognition were done with HTK.
- GNU Linear Programming Kit was used to solve the linear programming problems.

The original LME objective function is computationally intractable, so we approximate it with a linear function of individual LM parameters:

$$d(W|\Lambda) = \ln\left[\Pr(W_n|\Lambda) \cdot \mathcal{A}(W_n)\right] - \ln\sum_{W' \in \mathcal{G}_n} \left[\Pr(W'|\Lambda) \cdot \mathcal{A}(W')\right]$$

Reduction in Word and Sentence Error Rates

	Baseline	MMIE	LME
WER (%)			
Training Set	11.97	5.49 (54.14)	5.29 (55.81)
Test Set	27.00	26.38 (2.30)	26.30 (2.59)
SER (%)			
Training Set	25.6	12.20 (52.34)	11.6 (54.69)
Test Set	43.15	42.46 (1.60)	42.36 (1.83)







 λ_i, η_j, μ_k - tri-gram, bi-gram, uni-gram log-conditional probs. ϕ_l, ψ_m - bi-gram, uni-gram back-off weights in log-domain)

ve function as a standard LP $\mathbf{v} \left| \rho - \frac{\epsilon}{|\mathcal{E}|} \cdot c^T \Lambda \right|$ $C: \tilde{d}(W_n|\Lambda) \ge \rho$ $\Lambda : \lambda_i^{(n)} - \tau \le \lambda_i \le \lambda_i^{(n)} + \tau$ $\rho \ge 0$ $\ldots \sum e_1 \cdots \sum e_M$ $W {\in} \mathcal{E}$ $W \in \mathcal{E}$