Trans-dimensional Random Fields (TDRF) for Sequence Modeling

We present the potential of applying random fields for sequence modeling, demonstrated by its success in language modeling.

Bin Wang[1] ***Zhijian Ou***[1] ***Zhiqiang Tan***[2]

[1] **Dept. of Electronic Engineering, Tsinghua Univ., China**

[2] **Dept. of Statistics, Rutgers Univ., USA**

State-of-the-art LMs – Review

- Dominant: Conditional approach

$$p(x_1, x_2, \cdots, x_l) = \prod_{i=1}^{l} p(x_i | x_1, \cdots, x_{i-1})$$

- N-gram LMs
- Neural network LMs

$$p(x_i = k | x_1, \cdots, x_{i-1}) \approx \frac{w_k^T \phi[x_1, \cdots, x_{i-1}]}{\sum_{k=1}^{V} w_k^T \phi[x_1, \cdots, x_{i-1}]}, \ w_k \in R^h$$

☹ Computational expensive in both training and testing [1]
e.g. lexicon size $V = 10k \sim 100k$, embedding dim $h = 250$

[1] Partly alleviated by using un-normalized models, e.g. through noise contrastive estimation training.

TDRF LMs – Motivation

$p(x_1, x_2, \cdots, x_l) = ?$

Dominant:
Conditional approach / Directed

$$x_1 \to x_2 \to x_3 \to \cdots \to x_l$$

Alternative:
Random field approach / Undirected

$$x_1 - x_2 - x_3 - \cdots - x_l$$

☹ Model training is difficult.

☺ Capture bidirectional context for language cognition.

The cat is on the table.

The cat is in the house.

☺ **Breakthrough in training with a number of innovations**
Fixed-dim (e.g. image) -> Trans-dim (sequential modeling)

TDRF LMs – Model Definition

- Features ($f_i, i = 1,2,\dots,F$) can be defined flexibly.
- Each feature brings a contribution to the sentence probability.

$$p(x;\lambda) = \frac{1}{Z(\lambda)} \exp\left(\sum_{i=1}^{F} \lambda_i f_i(x)\right), x \triangleq (x_1, x_2, \cdots, x_l)$$

$$f_i(x) = \begin{cases} 1, & \text{'meeting on DAY–OF–WEEK' appears in } x \quad \Rightarrow \lambda_i \text{ is activated} \\ 0, & \text{Otherwise} \quad \Rightarrow \lambda_i \text{ is removed} \end{cases}$$

☺ More flexible features, beyond the n-gram features, can be well supported in TDRF LMs.

☺ Computational efficient in computing sentence probability for testing.

Jelinek 1995: put language back into language modeling

WSME vs TDRF

- Whole-sentence maximum entropy (WSME) (Rosenfeld, Chen, Zhu 2001)

$$p(l, x^l; \lambda) = \frac{1}{Z(\lambda)} \exp[\lambda^T f(x^l)], x \triangleq (l, x^l), x^l \triangleq (x_1, x_2, \cdots, x_l)$$

$$= \frac{Z_l(\lambda)}{Z(\lambda)} \cdot \frac{1}{Z_l(\lambda)} \cdot \exp[\lambda^T f(x^l)], \ Z_l(\lambda) = \sum_{x^l} \exp[\lambda^T f(x^l)]$$

A mixture distribution with unknown weights, which differ from each other greatly, e.g. 10^{40} !

Poor sampling → poor estimation of gradient → poor fitting

- Trans-dimensional RF (TDRF) model

$$p(l, x^l; \lambda) = \pi_l \cdot \frac{1}{Z_l(\lambda)} \cdot \exp[\lambda^T f(x^l)], \qquad l = 1, \cdots, m$$

Empirical length probabilities in the training data

Serve as a control device to improve sampling from multiple distributions!

TDRF LMs – Model Estimation

- Maximum-likelihood training

$$\frac{\partial LogLikelihood}{\partial \lambda} = E_{\tilde{p}(x)}[f_i(x)] - E_{p(x;\lambda)}[f_i(x)] = 0$$

Expectation under empirical distribution $\tilde{p}(x)$

Expectation under model distribution $p(x;\lambda)$

- Consider $p(l, x^l; \lambda, \zeta) \propto \pi_l \cdot \frac{1}{e^{\zeta_l}} \cdot \exp[\lambda^T f(x^l)]$

where ζ_l is hypothesized values of the true $\zeta_l^*(\lambda) = log Z_l(\lambda)$.

The marginal probability of length l is: $p(l; \lambda, \zeta) = \frac{\pi_l e^{-\zeta_l + \zeta_l^*(\lambda)}}{\sum_j \pi_l e^{-\zeta_j + \zeta_j^*(\lambda)}}$.

- Joint SA is used to find $\zeta_l^* = \zeta_l^*(\lambda^*)$ and λ^* that solves

$$\begin{cases} \pi_l = p(l; \lambda, \zeta), \qquad l = 1, \cdots, m \\ 0 = E_{\tilde{p}(x)}[f_i(x)] - E_{p(l, x^l; \lambda, \zeta)}[f_i(x)] \end{cases}$$

Experiments

LM Training — Penn Treebank portion of WSJ corpus

Test speech — WSJ'92 set, by rescoring of 1000-best lists

Type	Features
w	$(w_{-3}w_{-2}w_{-1}w_0)(w_{-2}w_{-1}w_0)(w_{-1}w_0)(w_0)$
c	$(c_{-3}c_{-2}c_{-1}c_0)(c_{-2}c_{-1}c_0)(c_{-1}c_0)(c_0)$
ws	$(w_{-3}w_0)(w_{-3}w_{-2}w_0)(w_{-3}w_{-1}w_0)(w_{-2}w_0)$
cs	$(c_{-3}c_0)(c_{-3}c_{-2}c_0)(c_{-3}c_{-1}c_0)(c_{-2}c_0)$
wsh	$(w_{-4}w_0)(w_{-5}w_0)$
csh	$(c_{-4}c_0)(c_{-5}c_0)$
cpw	$(c_{-3}c_{-2}c_{-1}w_0)(c_{-2}c_{-1}w_0)(c_{-1}w_0)$

model	WER	PPL (± std. dev.)	#feat
KN4	8.71	295.41	1.6M
RNN	7.96	256.15	5.1M
WSMEs (200c)			
w+c+ws+cs	8.87	$\approx 2.8 \times 10^{12}$	5.2M
w+c+ws+cs+cpw	8.82	$\approx 6.7 \times 10^{12}$	6.4M
TDRFs (100c)			
w+c	8.56	268.25±3.52	2.2M
w+c+ws+cs	8.16	265.81±4.30	4.5M
w+c+ws+cs+cpw	8.05	265.63±7.93	5.6M
w+c+ws+cs+wsh+csh	8.03	276.90±5.00	5.2M
TDRFs (200c)			
w+c	8.46	257.78±3.13	2.5M
w+c+ws+cs	8.05	257.80±4.29	5.2M
w+c+ws+cs+cpw	**7.92**	264.86±8.55	6.4M
w+c+ws+cs+wsh+csh	**7.94**	266.42±7.48	5.9M
TDRFs (500c)			
w+c	8.72	261.02±2.94	2.8M
w+c+ws+cs	8.29	266.34±6.13	5.9M

Comparison	Computation efficient in training	Computation efficient in testing	Bidirectional context	Flexible features	Performance
N-gram LMs	✔	✔	✘	✘	✘
Neural network LMs	✘	✘	✘	✔	✔
TDRF LMs	✘	✔	✔	✔	✔