概率图模型理论及应用

Theory and Applications of Probabilistic Graphical Models

(Lesson 5 - RFLearning)

欧智坚

清华大学电子工程系

Addr: 罗姆楼 6-104

Tel: 62796193

Email: ozj@tsinghua.edu.cn

课程章节

- 第一章 引言（1）
- 第二章 图模型的表示理论（2）
  - Semantics (DGM, UGM)
  - HMM, CRF
- 第三章 图模型的推理理论（6）
  - 精确推理：variable-elimination，cluster-tree，triangulate
  - 连续变量：Kalman
  - 采样近似：sampling
  - 变分近似：variational
- 第四章 图模型的学习理论（3）
  - 参数学习：maxlikelihoodEstimate，RFLearning, BayesEstimate
  - 结构学习：StructureLearning
- 第五章 一个综合例子（1）

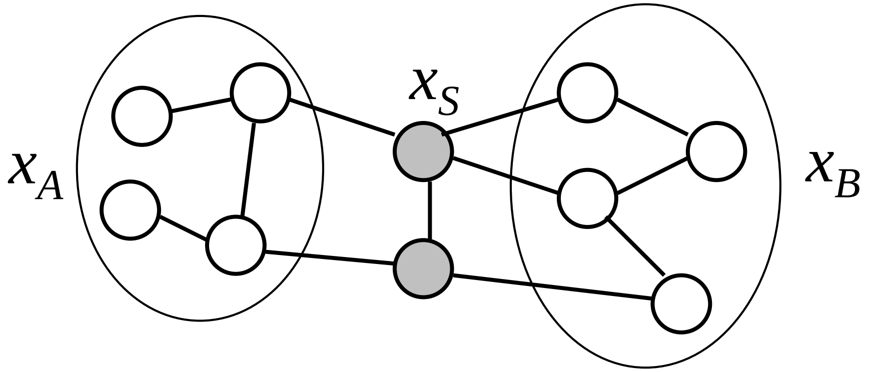
Introduction

- Probabilistic graphical models
  - A general framework for describing and applying statistical models
  - Statistical modeling, inference and learning
- Directed graphical models (DGMs)
  - aka Bayesian networks (BNs)
  - e.g. HMMs, Topic models (LDA)
- Undirected graphical models (UGMs)
  - aka Markov random fields (MRFs), random fields (RFs), Markov networks (MNs)
  - e.g. CRFs, RBMs, DBNs

UGM Semantics - (G) property

- A probability distribution $p(x_V)$ is said to obey the global Markov property, relative to g , if for any triple (A, B, S) of disjoint subsets of V such that S *separates A from B,*

$$x_A \perp x_B \mid x_S$$



S separates A from B : if all trails from A to B intersect S

UGM Semantics - Factorization property (F)

- A probability distribution $p(x_V)$ is said to factorize according to g, if there exist non-negative functions (called potential functions) $\phi_C(x_C)$ for all cliques C such that

$$p(x_V) = \frac{1}{Z}\prod_{C\in\mathcal{C}} \phi_C(x_C) \quad \text{or} \quad p(x_V) \propto \prod_{C\in\mathcal{C}} \phi_C(x_C)$$

where Z is the normalizing constant (partition function)

$$Z = \sum_{x_V}\prod_{C\in\mathcal{C}} \phi(x_C)$$

  - Potential functions $\phi_C(x_C)$ are not uniquely determined.
  - Without loss of generality, define potentials over maximum cliques.

Hammersley-Clifford Theorem: If p is strictly positive, (F)$\Longleftrightarrow$(G).

UGMs and Energy-based models

- Let every clique potential be associated with a clique energy $E(x_C)$

$$E_C(x_C) = -log\phi_C(x_C)$$

- The resulting joint is known as the Gibbs (or Boltzmann) distribution

$$p(x_V) \propto exp\left[-\sum_C E_C(x_C)\right]$$

High probability states correspond to low energy configurations.

UGMs and log-linear models

- Let each clique potential be a log-linear function

$$log\phi_C(x_C) = \theta_C^T f_C(x_C)$$

where $f_C(x_C)$ is a feature vector derived from the values of the variables x_C, θ_C is the associated feature weight vector.

- The resulting joint has the form

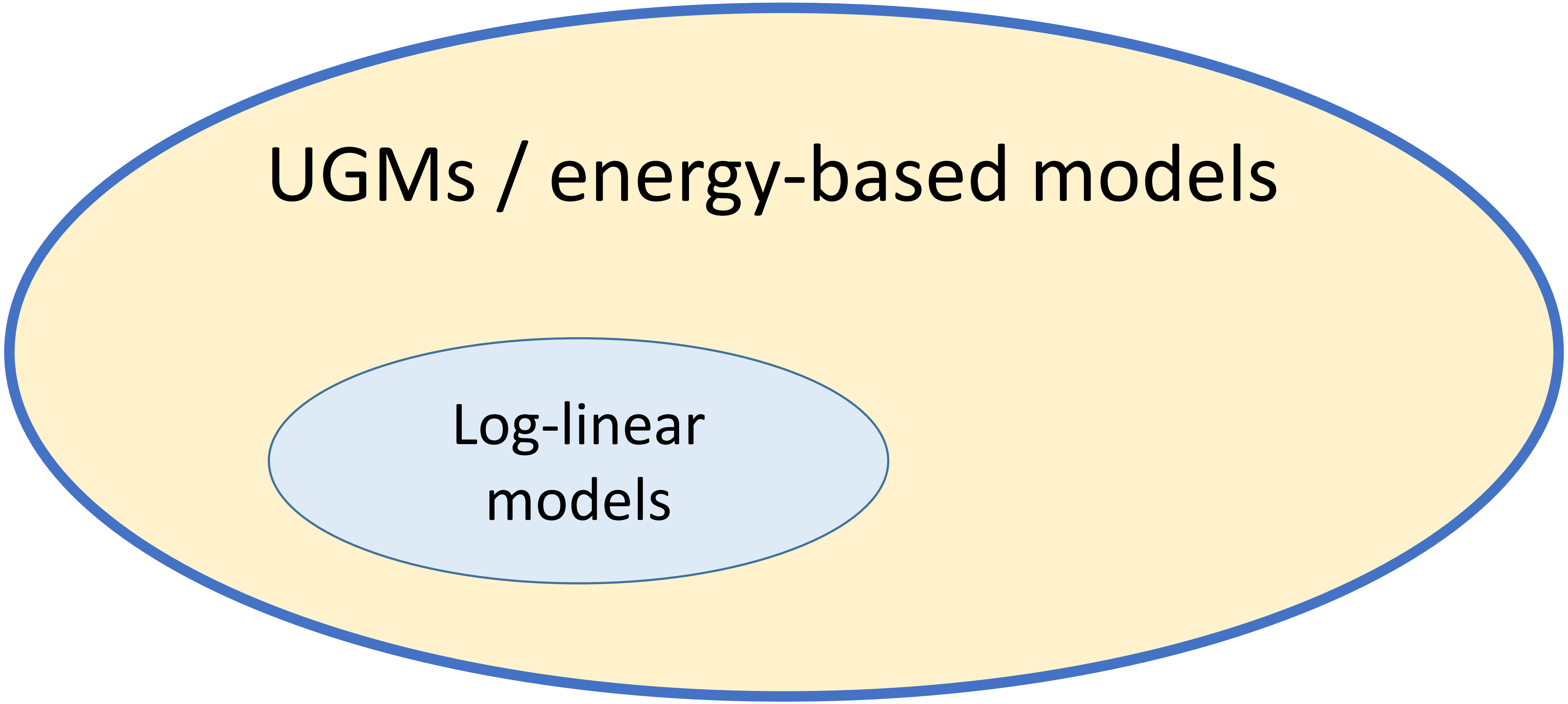
$$p(x_V) = \frac{1}{Z(\theta)} exp\left[\sum_C \theta_C^T f_C(x_C)\right]$$

This is known as a log-linear model or a Maximum Entropy model.

It can be proved that the maxent distribution is the same as
the maximum likelihood distribution from the closure of the set of log-linear RF distributions.

S. D. Pietra, V. D. Pietra, and J. Lafferty, "Inducing features of random fields", IEEE PAMI, 1997.

Relationship between UGMs and other models



Feature-based potential representation in log-linear models

- Consider an edge potential $\phi_{s,t}(x_s, x_t)$ associated with two discrete variables x_s and x_t, both of which can take K values.
- Define a feature vector of length K^2 as follows:

$$f_{s,t}(x_s, x_t) = [\cdots, 1(x_s = j, x_t = k), \cdots]^T, \qquad j, k = 1, \cdots, K$$

  with the associated weights:

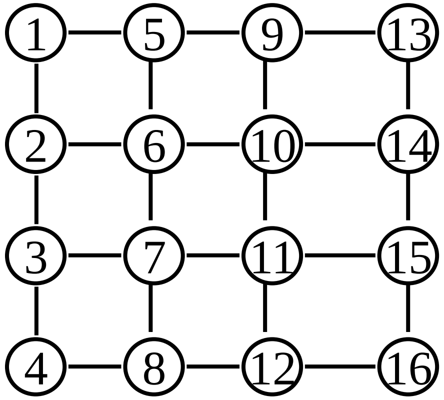
$$\theta_{s,t} = \left[\cdots, log\left(\phi_{s,t}(x_s = j, x_t = k)\right), \cdots\right]^T, \qquad j, k = 1, \cdots, K$$

- Then the tabular potential $\phi_{s,t}(x_s, x_t)$ can be represented as the log-liner form

$$\phi_{s,t}(x_s, x_t) = exp[\theta_{s,t}^T f_{s,t}(x_s, x_t)]$$

- Note: the log-linear form is more general because we can choose (or learn) the features.

UGM Example - Ising model



- Consider a lattice of binary RV's, $x_i \in \{-1,1\}$

$$p\left(x_{1:N^2}\right) \propto \exp\left\{\sum_{(i,j)\in E} \psi\left(x_i, x_j\right)\right\} = \exp\left\{\beta \sum_{(i,j)\in E} x_i x_j\right\} \quad \beta > 0$$

  - β : how much neighboring variables take identical values is favored.
  - Samples of Ising models on a lattice with different β:

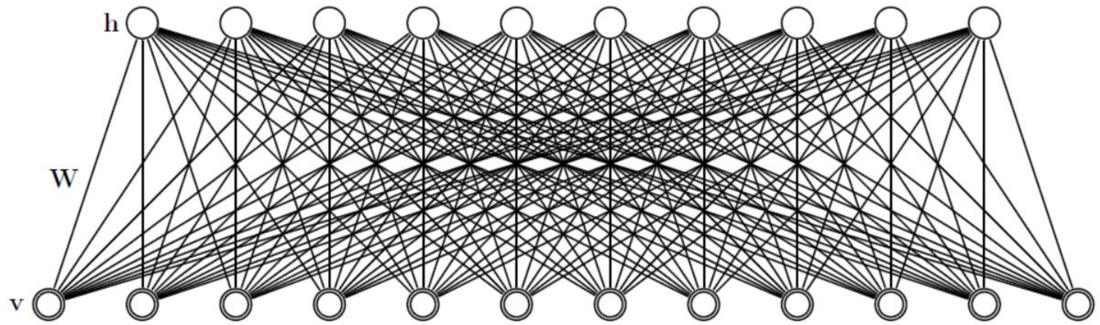
β = 0.1 or 10 ?

$\beta = 1$

β = 0.1 or 10 ?

Restricted Boltzmann Machines (RBMs)

- RBM is the main building block of a Deep Belief Network
- RBM is a two-layer MRF
  - Binary visible variables $v \in \{0,1\}^D$
  - Binary hidden variables $h \in \{0,1\}^F$
  - $\theta = \{W, b, a\}$



RBM: a stochastic version of a NN

$$p(v,h;\theta) = \frac{1}{Z(\theta)} exp[-E(v,h;\theta)]$$

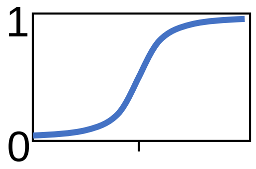
$$E(v,h;\theta) = -v^T W h - b^T v - a^T h$$

$$= -\sum_{i=1}^{D}\sum_{j=1}^{F} v_i W_{ij} h_j - \sum_{i=1}^{D} b_i v_i - \sum_{j=1}^{F} a_j h_j$$

$$p(h|v;\theta) = \prod_j p(h_j|v), \qquad p(h_j = 1|v) = \sigma\left(\sum_i W_{ij} v_i + a_j\right)$$

$$p(v|h;\theta) = \prod_i p(v_i|h), \qquad p(v_i = 1|h) = \sigma\left(\sum_j W_{ij} h_j + b_i\right)$$

Sigmoid function : $\sigma(x) = \frac{1}{1+e^{-x}}$

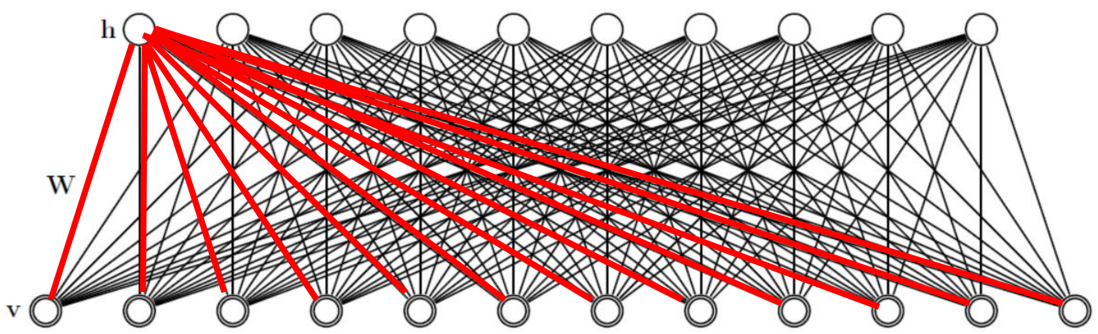


Ruslan Salakhutdinov. “Learning deep generative models”. PhD thesis, University of Toronto, 2009.

Learned features W_{*j}

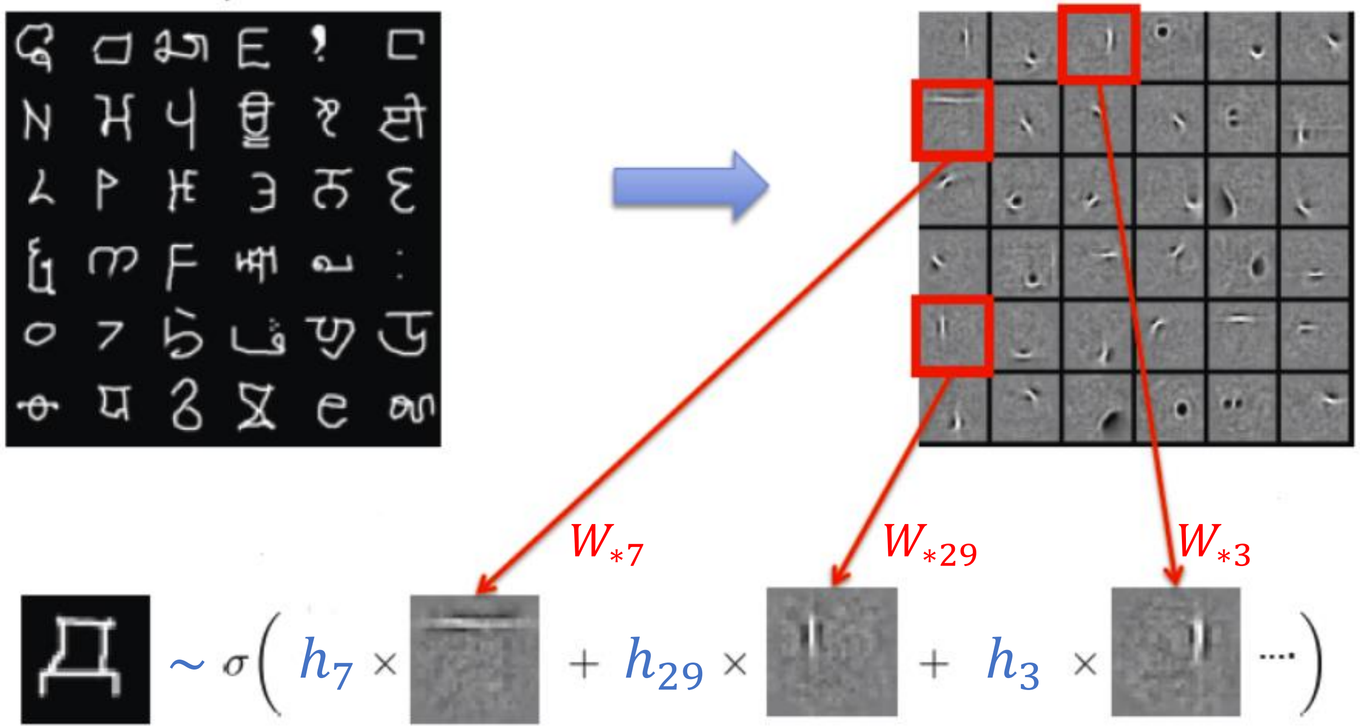
Learned receptive fields for unit h_j

h: higher-level encoding of v



Observed Data
Subset of 25,000 characters

Learned W: "edges"/"parts"
Subset of 1000 features



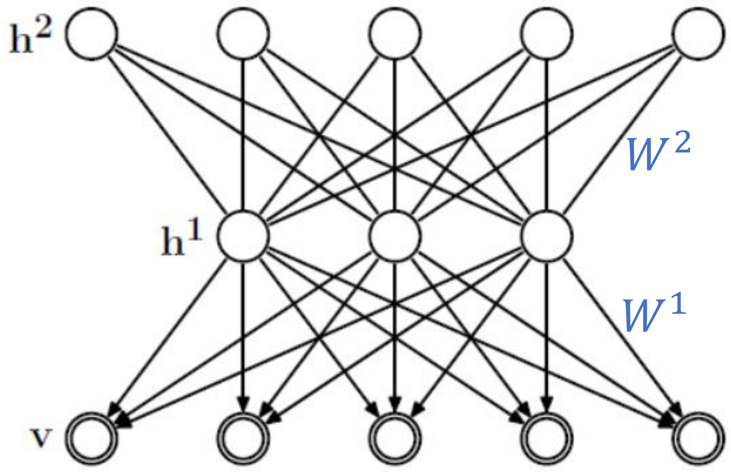
$$p(v|h;\theta) = \prod_i p(v_i|h),$$

$$p(v_i = 1|h) = \sigma\left(\sum_j W_{ij}h_j + b_i\right)$$

$$v|h \sim \sigma(h_1 \cdot W_{*1} + h_2 \cdot W_{*2} + \cdots + b)$$

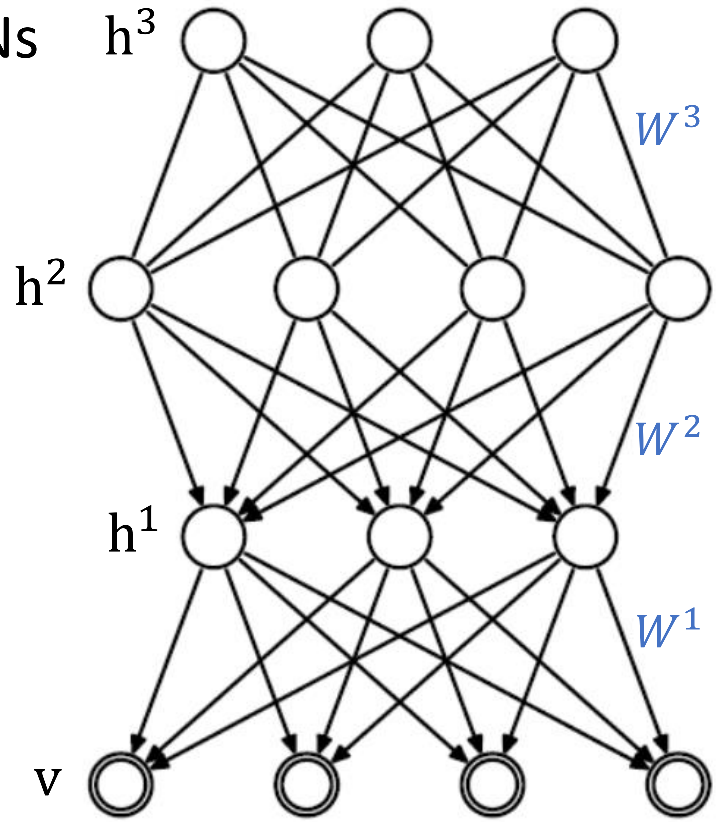
Deep Belief Networks (DBNs)

- DBNs ignite Deep Learning, Science 2006
- DBN is a multilayer mixed directed and undirected model
  - Greedy layer-by-layer learning as pre-training for DNNs



$$p(v, h^1, h^2; \theta) = p(v|h^1; W^1)p(h^1, h^2; W^2)$$

$$\theta = \{W^1, W^2\}$$

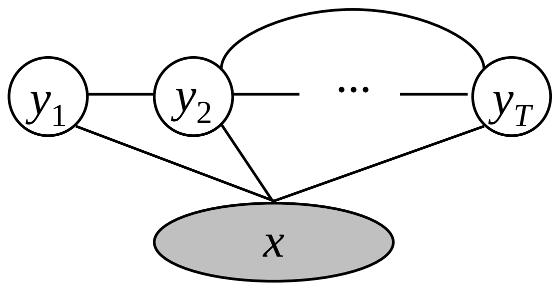


$$p(v, h^1, h^2, h^3; \theta) = p(v|h^1; W^1)p(h^1|h^2; W^2)p(h^2, h^3; W^3)$$

$$\theta = \{W^1, W^2, W^3\}$$

Conditional Random Fields (CRFs)

- A CRF is a conditional distribution defined as a MRF



$$p(y|x) = \frac{1}{Z(x)} exp\left[\sum_{C} \psi_C(y_C, x)\right]$$

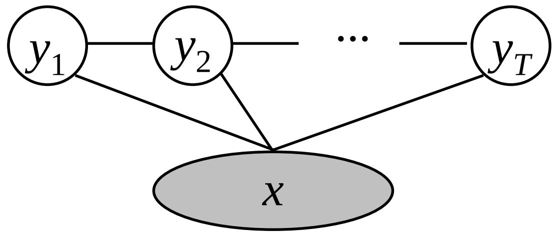
- x is observed sequence, which is always given;
- y is hidden sequence;
- $\psi_C(y_C, x)$: Clique feature function.

J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data", ICML 2001.

Linear-chain CRFs

for sequence tagging, e.g. POS tagging, shallow parser, Chinese word segmentation, …

$$p\left(y_{1:T} \mid x\right) \propto \exp\left\{\sum_{t=1}^{T-1} \psi_t\left(y_t, y_{t+1}, x\right) + \sum_{t=1}^{T} \psi_t\left(y_t, x\right)\right\}$$



Log-linear representation of tabular potentials

$$p\left(y_{1:T} \mid x\right) \propto \exp\left\{\sum_{t=1}^{T-1} \sum_i \lambda_i f_i\left(y_t, y_{t+1}, x, t\right) + \sum_{t=1}^{T} \sum_j \mu_j f_j\left(y_t, x, t\right)\right\}$$

Transition/edge features

$$\lambda_i f_i\left(y_t, y_{t+1}, x, t\right) = \lambda_i \cdot 1\left(y_t = prep, y_{t+1} = non\right)$$

State/node features

$$\mu_j f_j\left(y_t, x, t\right) = \mu_j \cdot 1\left(y_t = prep,\ x_t = \text{on}\right)$$

$$\mu_j f_j\left(y_t, x, t\right) = \mu_j \cdot 1\left(y_t = adv,\ x_t \text{ ends in } ly\right)$$

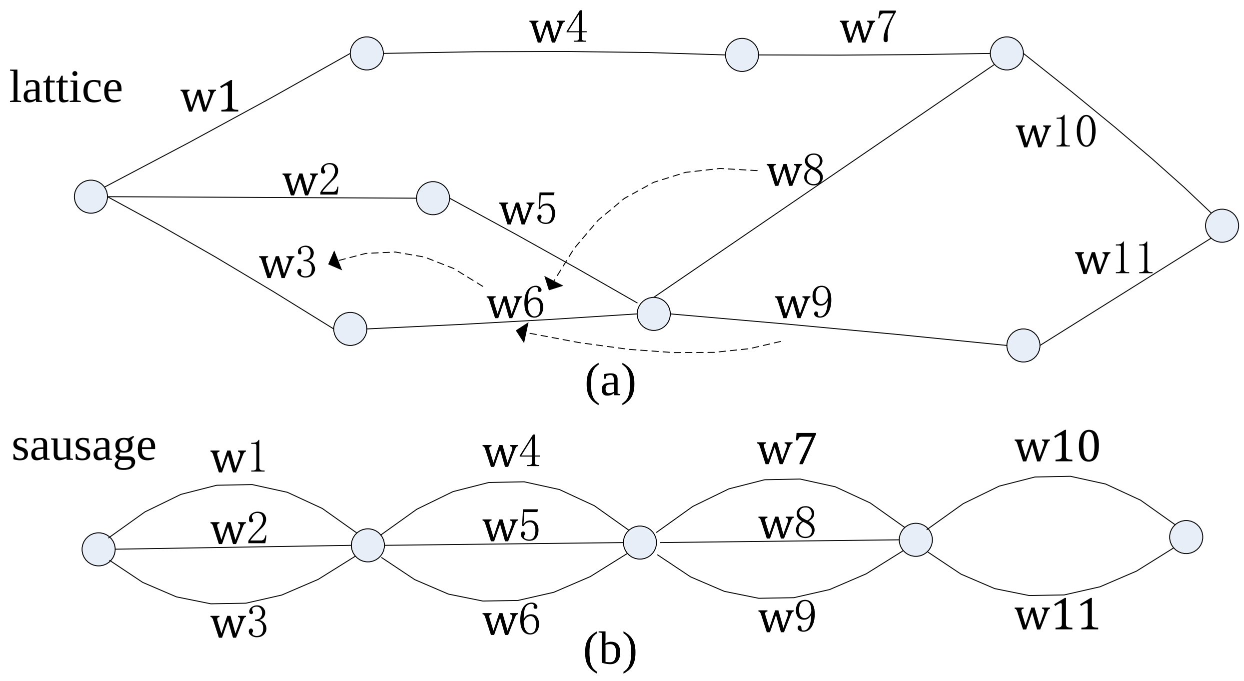
Why UGMs ?

$$p(x_V) \propto \prod_{v \in V} p\left(x_v \mid x_{pa(v)}\right) \quad \text{vs} \quad p(x_V) = \frac{1}{Z}\prod_{C \in \mathcal{C}} \phi_C(x_C)$$

- Advantages over DGMs
  - Undirected modeling is more natural for co-occurrence, where fixing the directions of edges is awkward in a graphical model.
  - Avoid local normalization and acyclicity requirements
    - Potentially more powerful modeling capacity
      - e.g. CRFs overcome the label bias weakness.
    - Easily encode a much richer set of patterns/features
- Disadvantages over DGMs
  - Parameter learning in UGMs may be more computational expensive.
- The inference problem is (basically) the same in DGMs and UGMs.
  - UGMs are computational more efficient by avoiding softmax calculation

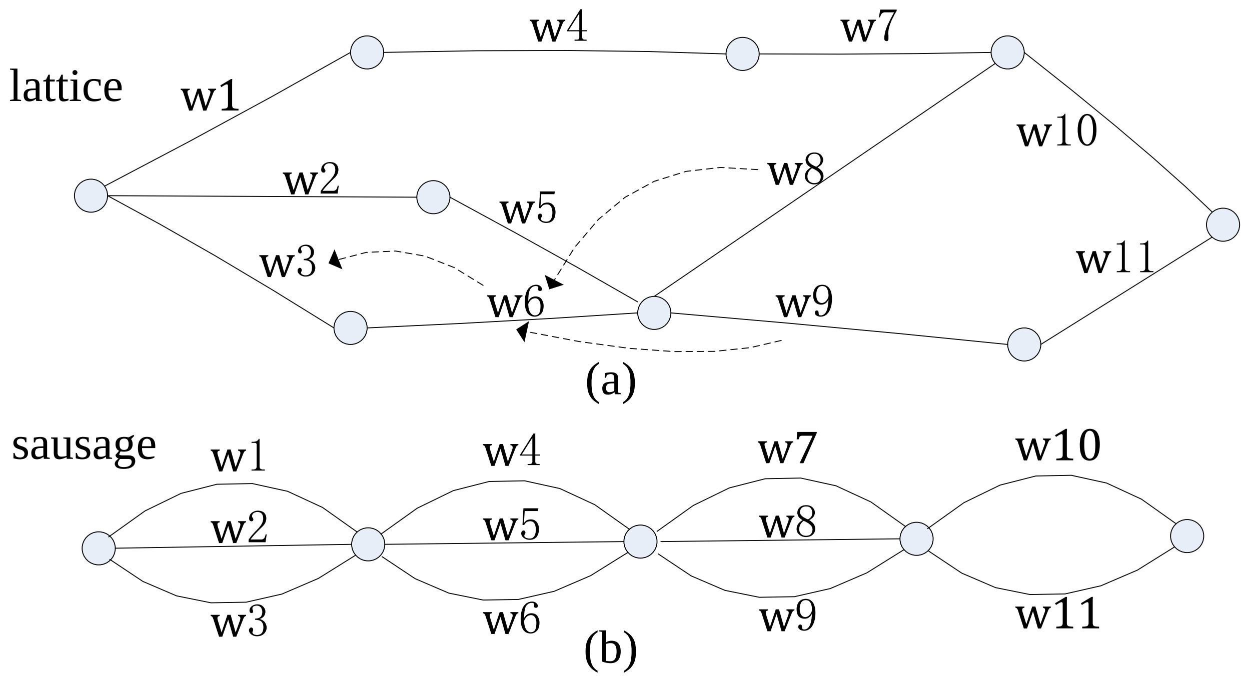
Case study: CRF-based confidence measure (CM)

- Motivation
  - The use of forward-backward posterior probabilities as the confidence scores
  - Limitation: its performance for CMs cannot be improved easily.
  - Use CRFs to combine various relevant features !



1. Reduce lattice to sausage (a linear sequence of slices**) so that (linear-chain) CRFs can be used.**

Case study: CRF-based confidence measure (CM)



2. Define the CRF over sausage

Given the sausage y, the reliability of the word candidate w4 is $p(q_2\text{=w4} \mid y)$.

$$p(q|y) \propto exp\left\{\sum_{n=1}^{N} \phi_n(q_n, y) + \sum_{n=2}^{N} \psi_n(q_{n-1}, q_n, y)\right\}$$

Z. Ou, H. Luo. "CRF-based Confidence Measures of Recognized Candidates for Lattice-based Audio Indexing." ICASSP 2012.

Case study

Trans-dimensional Random Field Language Models (TRF LMs) – brand new

- State-of-the-art LMs - review
  - N-gram LMs
  - Neural network LMs
- Motivation - why
- Model formulation - what
- Model Training - breakthrough
- Experiment results - evaluation
- Summary

N-gram LMs

- Language modeling (LM) is to determine the joint probability of a sentence, i.e. a word sequence.
- Dominant: Conditional approach

Current word — All previous words/history

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

Previous $n-1$ words

$$\approx \prod_{i=1}^{l} \underline{p(x_i | x_{i-n+1}, \cdots, x_{i-1})}$$

- Using Markov assumption leads to the N-gram LMs
  – One of the state-of-the-art LMs

Neural network LMs

- Another state-of-the-art LMs

history

$x_1, \cdots, x_{i-1}$ → Neural Network → $\phi[x_1, \cdots, x_{i-1}] \triangleq \phi \in R^h$

$$p(x_i | x_1, \cdots, x_{i-1}) \approx p(x_i | \phi[x_1, \cdots, x_{i-1}])$$

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

☹ Computational very expensive in both training and testing [1]

e.g. $V = 10k{\sim}100k, h = 250$

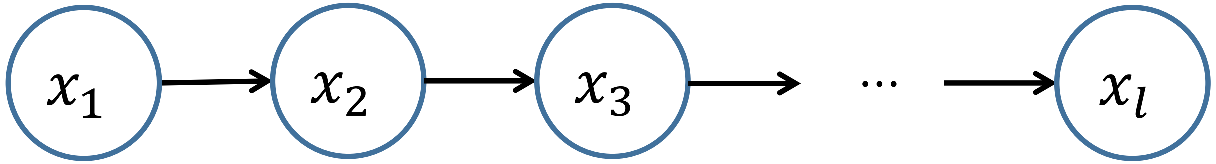
[1] Partly alleviated by using un-normalized models, e.g. through noise contrastive estimation training.

TRF LMs – Motivation (1)

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

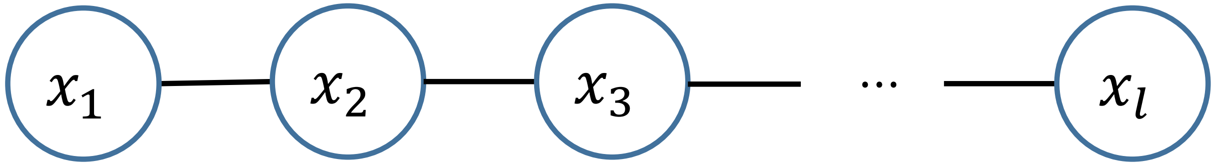
Dominant:

Conditional approach / Directed



Alternative:

Random field approach / Undirected



☹ Difficulty in model training

☺ A rule in language cognition: employ context for reading and writing

The cat is on the table.

The cat is in the house.

TRF LMs – Motivation (2)

- Drawback of N-gram LMs
  - N-gram is only one type of linguistic feature/property/constraint
  - meeting on Monday

$$P(w_i = Monday|w_{i-2} = meeing, w_{i-1} = on)$$

  - What if the training data only contain ‘meeting on Monday’ ?
  - New feature ‘meeting on DAY-OF-WEEK’, using class
  - New feature ‘party on *** birthday’, using skip
  - New features ….

F. Jelinek, 1932 – 2010

- 1985: Every time I fire a linguist, the performance of the speech recognizer goes up.
- 1995: put language back into language modeling.

TRF LMs – Formulation

- Intuitive idea
  - Features ($f_i, i = 1,2,\dots,F$) can be defined flexibly, beyond the n-gram features.
  - Each feature brings a contribution to the sentence probability $p(x)$
- Formulation

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

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

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

☺ Computational very efficient in computing sentence probability.

TRF LMs – Breakthrough in training (1)

- Propose Joint Stochastic Approximation (SA) Training Algorithm
  - Simultaneously updates the model parameters and normalization constants

Algorithm 1 Joint stochastic approximation

Input: training set

set initial values $\lambda^{(0)} = (0, \ldots, 0)^T$ and
$\zeta^{(0)} = \zeta^*(\lambda^{(0)}) - \zeta_1^*(\lambda^{(0)})$

for $t = 1, 2, \ldots, t_{max}$ **do**

  set $B^{(t)} = \emptyset$

  set $(L^{(t,0)}, X^{(t,0)}) = (L^{(t-1,K)}, X^{(t-1,K)})$

  Step I: MCMC sampling

  for $k = 1 \rightarrow K$ **do**

    sampling (See Algorithm 3)
$(L^{(t,k)}, X^{(t,k)}) = SAMPLE(L^{(t,k-1)}, X^{(t,k-1)})$

    set $B^{(t)} = B^{(t)} \cup \{(L^{(t,k)}, X^{(t,k)})\}$

  end for

  Step II: SA updating

  Compute $\lambda^{(t)}$ based on (13)

  Compute $\zeta^{(t)}$ based on (14) and (15)

end for

TRF LMs – Breakthrough in training (2)

- Propose Trans-dimensional mixture sampling
  - Sampling from $p(l, x^l; \lambda, \zeta)$, a mixture of RFs on subspaces of different dimensions.
  - Formally like RJ-MCMC (Green, 1995).

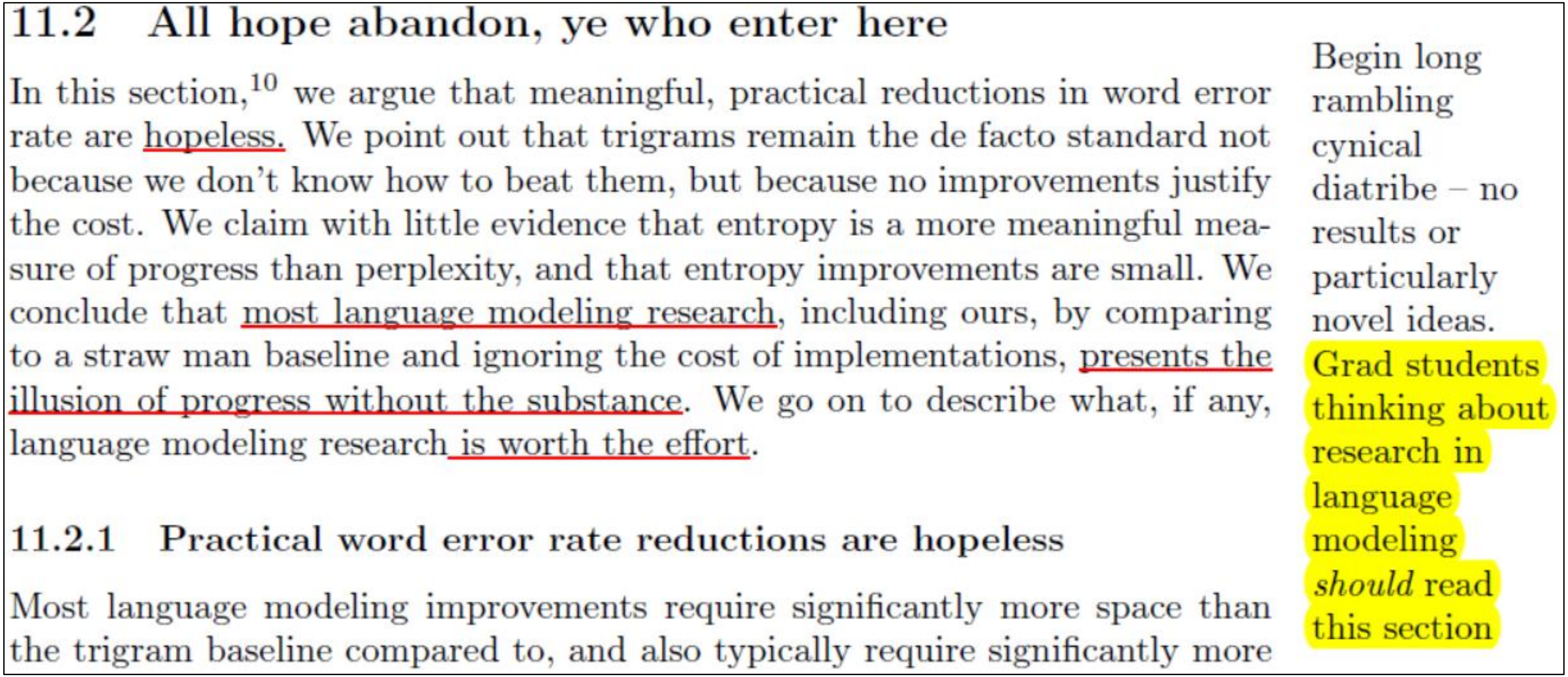
function SAMPLING$((L^{(t-1)}, X^{(t-1)}))$
  set $k = L^{(t-1)}$
  set $L^{(t)} = k$
  set $X^{(t)} = X^{(t-1)}$
  Stage I: Local jump
  generate $j \sim \Gamma(k, \cdot)$
  if $j = k + 1$ **then**
    generate $Y \sim g_{k+1}(y|X^{(t-1)})$ (equ.24)
    set $L^{(t)} = j$ and $X^{(t)} = \{X^{(t-1)}, Y\}$ with probability equ.22
  end if
  if $j = k - 1$ **then**
    set $L^{(t)} = j$ and $X^{(t)} = X^{(t-1)}_{1:k-1}$ with probability equ.23
  end if
  Stage II: Markov move
  for $i = 1 \rightarrow L^{(t)}$ **do**
    $a \sim p(L^{(t)}, \{X^{(t)}_{1:i-1}, \cdot, X^{(t)}_{i+1:L^{(t)}}\}; \Lambda, \zeta)$
    $X^{(t)}_i \leftarrow a$
  end for
  return $(L^{(t)}, X^{(t)})$
end function

Experiment results

- Benchmarking experiments
  - Speech recognition on PTB-WSJ dataset
  - Speech recognition on ChiME-4 dataset
  - Mandarin speech recognition on Toshiba dataset
- TRF LMs significantly outperform KN n-gram LMs (10%+ WER relative reduction), and perform better than RNN LMs and close to LSTM LMs but with much faster speed in computing sentence probabilities (0.16 sec. CPU vs 40 sec. GPU).
- Interpolated TRF and LSTM is better than Interpolated KN5 and LSTM.

- Bin Wang, Zhijian Ou, Zhiqiang Tan, "Trans-dimensional Random Fields for Language Modeling", ACL 2015.
- Bin Wang, Zhijian Ou, Yong He, and Akinori Kawamura, "Model Interpolation with Trans-dimensional Random Field Language Models for Speech Recognition", *arXiv* 2016.
- Hongyu Xiang, Bin Wang and Zhijian Ou. "The THU-SPMI CHiME-4 system : Lightweight design with advanced multi-channel processing, feature enhancement, and language modeling". CHiME Workshop, 2016,9.

Once said in : J. Goodman, "A bit of progress in language modeling", Computer Speech & Language, 2001.



11.2 All hope abandon, ye who enter here

In this section,[10] we argue that meaningful, practical reductions in word error rate are hopeless. We point out that trigrams remain the de facto standard not because we don't know how to beat them, but because no improvements justify the cost. We claim with little evidence that entropy is a more meaningful measure of progress than perplexity, and that entropy improvements are small. We conclude that most language modeling research, including ours, by comparing to a straw man baseline and ignoring the cost of implementations, presents the illusion of progress without the substance. We go on to describe what, if any, language modeling research is worth the effort.

11.2.1 Practical word error rate reductions are hopeless

Most language modeling improvements require significantly more space than the trigram baseline compared to, and also typically require significantly more

Begin long rambling cynical diatribe – no results or particularly novel ideas. Grad students thinking about research in language modeling *should* read this section

Now we can beat n-gram significantly by RNN (Mikolov, 2010) and TRF (2015) …

Contents

Training of UGMs in general

$$p(x;\theta) = \frac{1}{Z(\theta)}\exp[Q(x;\theta)]$$

Normalization constant:

$$Z(\theta) = \sum_x \exp[Q(x;\theta)]$$

- Maximum likelihood (ML) training

The scaled log-likelihood of observed $\{x_i, i = 1, \cdots, N\}$

$$l(\theta) \triangleq \frac{1}{N}\sum_{i=1}^{N} logp(x_i;\theta) = \left[\frac{1}{N}\sum_{i=1}^{N} Q(x_i;\theta)\right] - logZ(\theta)$$

$$\frac{\partial l(\theta)}{\partial\theta} = E_{\tilde{p}(x)}\left[\frac{\partial Q(x;\theta)}{\partial\theta}\right] - E_{p(x;\theta)}\left[\frac{\partial Q(x;\theta)}{\partial\theta}\right] = 0$$

Maximum Entropy

Expectation under empirical distribution $\tilde{p}(x) = \frac{1}{N}\sum_{i=1}^{N} 1(x = x_i)$

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

Training of UGMs - overview

- Roughly speaking, two types of approximate methods
- Gradient methods
  - Make explicit use of the gradient: Gradient descent, conjugate gradient, L-BFGS.
  - Stochastic approximation (SA)
  - Stochastic maximum likelihood (SML)
  - Persistent contrastive divergence (PCD)
- Lower bound methods
  - Generalized iterative scaling (GIS)
  - Improved iterative scaling (IIS)
  - Mostly studied in the context of maximum entropy (maxent) parameter estimation of log-linear models.
- In practice the gradient methods are shown to be much faster than the lower bound methods

Comparison on learning CRFs

- Div: the relative entropy between the fitted model and the training data
- Iter: Iteration number
- Evals: the number of calculating log-likelihood and gradient
- Time: the total time.

Dataset	Method	Div.	Iter	Evals	Time (secs)
rules	gis	5.19×10^{-2}	1201	1202	23.04
	iis	5.14×10^{-2}	923	924	42.48
	steepest ascent	5.13×10^{-2}	212	331	6.16
	conjugate gradient (fr)	5.07×10^{-2}	74	196	3.74
	conjugate gradient (prp)	5.08×10^{-2}	63	154	2.87
	limited memory variable metric	5.07×10^{-2}	70	76	1.44
lex	gis	1.61×10^{-3}	370	371	36.29
	iis	1.52×10^{-3}	241	242	102.18
	steepest ascent	3.47×10^{-3}	1041	1641	139.10
	conjugate gradient (fr)	1.39×10^{-3}	166	453	39.03
	conjugate gradient (prp)	1.62×10^{-3}	150	382	32.46
	limited memory variable metric	1.49×10^{-3}	136	143	17.25
summary	gis	1.83×10^{-3}	1446	1447	125.46
	iis	1.07×10^{-3}	626	627	208.22
	steepest ascent	2.64×10^{-3}	1163	3503	227.30
	conjugate gradient (fr)	1.01×10^{-4}	175	948	60.91
	conjugate gradient (prp)	7.30×10^{-4}	93	428	27.81
	limited memory variable metric	3.98×10^{-5}	81	89	10.38
shallow	gis	3.57×10^{-2}	3428	3429	27103.62
	iis	3.50×10^{-2}	3216	3217	71053.24
	steepest ascent †	—	—	—	—
	conjugate gradient (fr)	2.91×10^{-2}	1094	6056	46958.87
	conjugate gradient (prp)	4.13×10^{-2}	421	2170	16477.84
	limited memory variable metric	3.26×10^{-2}	429	444	3408.30

- T. Tieleman, "Training restricted boltzmann machines using approximations to the likelihood gradient", ICML 2008.
- R. Malouf, "A comparison of algorithms for maximum entropy parameter estimation", in Proc. Conference on Natural Language Learning (CoNLL), 2002.

Training of log-linear models

$$p(x;\theta) = \frac{1}{Z(\theta)} exp\left[\sum_C \theta_C^T f_C(x)\right]$$ where C indexes the cliques.

$$\frac{\partial l(\theta)}{\partial \theta_C} = E_{\tilde{p}(x)}[f_C(x)] - E_{p(x;\theta)}[f_C(x)] = 0$$ Statistics matching

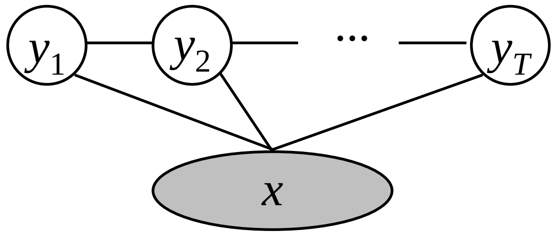
Empirical statistics of features

Expected statistics of features

- $l(\theta)$ is convex in θ, so it has a unique global maximum which we can find using gradient-based optimizers. ☺
- The exact calculation of the gradient is intractable in general, involving high-dimensional integration. ☹

Training of log-linear models - example

$$p(y_{1:T} \mid x) \propto \exp\left\{\sum_{t=1}^{T-1}\sum_i \lambda_i f_i(y_t, y_{t+1}, x, t) + \sum_{t=1}^{T}\sum_j \mu_j f_j(y_t, x, t)\right\}$$



- Maximum conditional likelihood (MCL)

$$\frac{\partial p(y_{1:T}|x;\theta)}{\partial \mu_j} = \sum_{t=1}^{T} f_j(y_t, x, t) - E_{p(y|x;\theta)}\left[\sum_{t=1}^{T} f_j(y_t, x, t)\right]$$

Statistics matching

$$= \sum_{t=1}^{T} f_j(y_t, x, t) - \sum_{t=1}^{T} E_{p(y_t|x;\theta)}[f_j(y_t, x, t)]$$

- The above gradient involves only one training instance $y_{1:T}|x$.
- The gradient of scaled conditional likelihood is sum of gradients for all training instances.

Training of partially observed UGMs

$$p(x,h;\theta) = \frac{1}{Z(\theta)} \exp[Q(x,h;\theta)]$$

Normalization constant:

$$Z(\theta) = \sum_{x,h} \exp[Q(x,h;\theta)]$$

- Maximum likelihood (ML) training

Scaled log-likelihood of observed $\{x_i, i = 1, \cdots, N\}$

$$l(\theta) \triangleq \frac{1}{N}\sum_{i=1}^{N} log p(x_i;\theta) = \left[\frac{1}{N}\sum_{i=1}^{N} log \sum_{h} \exp[Q(x_i,h;\theta)]\right] - log Z(\theta)$$

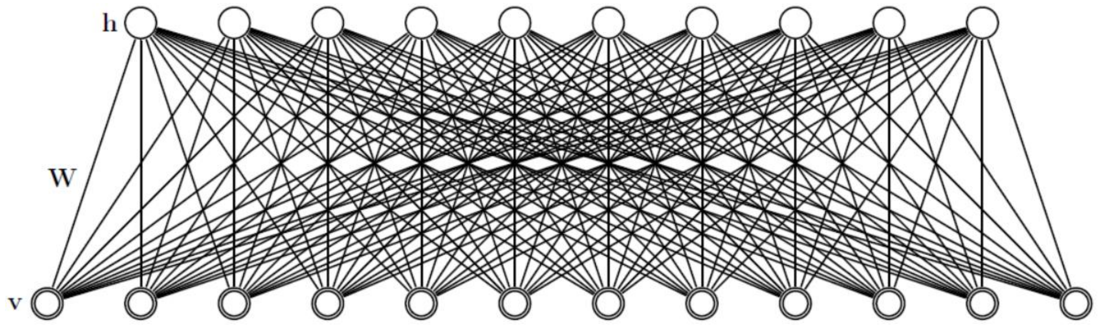
$$\frac{\partial l(\theta)}{\partial \theta} = E_{\tilde{p}(x)p(h|x)}\left[\frac{\partial Q(x,h;\theta)}{\partial \theta}\right] - E_{p(x,h;\theta)}\left[\frac{\partial Q(x,h;\theta)}{\partial \theta}\right] = 0$$

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

Expectation under model distribution $p(x,h;\theta)$

Training of partially observed UGMs - example

- RBM is a two-layer MRF
  - Binary visible variables $v \in \{0,1\}^D$
  - Binary hidden variables $h \in \{0,1\}^F$
  - $\theta = \{W, b, a\}$



$$p(v,h;\theta) = \frac{1}{Z(\theta)} exp[Q(v,h;\theta)]$$

$$Q(v,h;\theta) = v^T W h + b^T v + a^T h$$

$$= \sum_{i=1}^{D}\sum_{j=1}^{F} v_i W_{ij} h_j + \sum_{i=1}^{D} b_i v_i + \sum_{j=1}^{F} a_j h_j$$

$$\begin{cases} \frac{\partial l(\theta)}{\partial W} = E_{p_{emp}}[vh^T] - E_{p_{model}}[vh^T] \\ \frac{\partial l(\theta)}{\partial a} = E_{p_{emp}}[h] - E_{p_{model}}[h] \\ \frac{\partial l(\theta)}{\partial b} = E_{p_{emp}}[v] - E_{p_{model}}[v] \end{cases}$$

Training of UGMs in general

gradient = empirical expectation – model expectation

1. Approximate the model expectations using Monte Carlo sampling.
   - We can use MCMC to generate the samples, but running MCMC to convergence at each step of the inner loop would be extremely slow.
   - Fortunately, it was shown by Younes (1989) that we can start the MCMC chain at its previous value, and just take a few steps.
2. We can combine this with stochastic gradient descent (SGD), which takes samples from the empirical distribution.

Both two ideas/tricks essentially follows in the framework of Stochastic Approximation (SA).

- Robbins and Monro (1951). A stochastic approximation method. Ann. Math. Stat.
- L. Younes, “Parametric inference for imperfectly observed gibbsian fields,” Probability Theory and Related Fields, 1989.

Stochastic Approximation (SA)

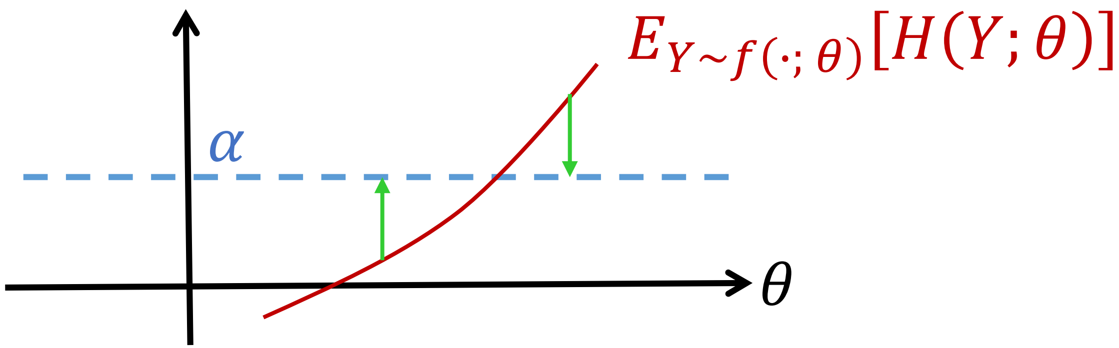
Problem: The objective is to find a solution θ to $E_{Y\sim f(\cdot;\,\theta)}[H(Y;\theta)] = \alpha$, where $\theta \in R^d$, noisy observation $H(Y;\theta) \in R^d$

Method:

(1) Sampling: Generate $Y_t \sim K(Y_{t-1}, \cdot\,; \theta_{t-1})$, a Markov transition kernel that admits $f(\cdot;\ \theta_{t-1})$ as the invariant distribution.

(2) Updating: Set $\theta_t = \theta_{t-1} + \gamma_t\{H(Y_t;\theta_{t-1}) - \alpha\}$

$$e.g.\ \gamma_t = \frac{1}{t_0 + t}$$



- Robbins and Monro (1951). A stochastic approximation method. Ann. Math. Stat.
- Chen (2002), Stochastic Approximation and Its Applications, Kluwer Academic Publishers.

Training of partially observed UGMs – SA algorithm

Training data : Observed $\{x_i, i = 1, \cdots, N\}$

$$\frac{\partial l(\theta)}{\partial \theta} = E_{\tilde{p}(v,z;\theta)}\left[\frac{\partial Q(v,z;\theta)}{\partial \theta}\right] - E_{p(x,h;\theta)}\left[\frac{\partial Q(x,h;\theta)}{\partial \theta}\right] = 0$$

$$Y = \begin{pmatrix} v \\ z \\ x \\ h \end{pmatrix} \sim f(\cdot;\ \theta) = \tilde{p}(v,z;\theta)p(x,h;\theta) \qquad \tilde{p}(v,z) = \frac{1}{N}\sum_{i=1}^{N} 1(v = x_i) \cdot p(z|v)$$

(1) Initialize θ_0 randomly;

(2) For iteration $t = 1, \cdots,$ do

- Draw a empirical minibatch of size B $\left\{\left(v^{(i)}, z^{(i)}\right), i = 1, \cdots, B\right\}$ according to $\tilde{p}(v, z; \theta_{t-1})$;

  Draw a Monte Carlo minibatch $\left\{\left(x^{(i)}, h^{(i)}\right), i = 1, \cdots, B\right\}$ by continuously taking B steps using a Markov transition kernel that admits $p(x, h; \theta_{t-1})$ as the invariant distribution.
- Updating:

$$\theta_t = \theta_{t-1} + \gamma_t \frac{1}{B}\left\{\left(\sum_{i=1}^{B} \frac{\partial Q\left(v^{(i)}, z^{(i)}; \theta\right)}{\partial \theta}\right)\Bigg|_{\theta=\theta_{t-1}} - \left(\sum_{i=1}^{B} \frac{\partial Q\left(x^{(i)}, h^{(i)}; \theta\right)}{\partial \theta}\right)\Bigg|_{\theta=\theta_{t-1}}\right\}$$

Connection of SA with other gradient methods

- Robbins and Monro 1951.
- aka Stochastic Maximum Likelihood (SML), (Younes 1989).
- This was independently discovered by Tieleman in 2008, who called it persistent contrastive divergence (PCD).
- In regular contrastive divergence (CD), proposed by Hinton 2002, we restart the Markov chain at the training data rather than at the previous state. This will not converge to the MLE.
- “Clearly, the widely used practice of CD1 learning is a rather poor “substitute” for maximum likelihood learning. “ (Salakhutdinov phd thesis 2009).

GIS and IIS for learning log-linear models

$$p(x;\lambda) = \frac{1}{Z}\exp\left\{\sum_{i=1}^{F}\lambda_i f_i(x)\right\}$$

$$\tilde{p}[f_i] - \sum_{x} p(x) f_i(x) e^{\Delta\lambda_i f_{\#}(x)} = 0 \qquad f_{\#}(x) = \sum_{i=1}^{F} f_i(x)$$

- Generalized iterative scaling (GIS)
  - Introduce an extra feature $f_{F+1}(x) = S - \sum_{i=1}^{F} f_i(x)$
  - Then we have $f_{\#}(x) = \sum_{i=1}^{F+1} f_i(x) = S$ is a constant.

$$\Delta\lambda_i = \frac{1}{S}\log\frac{\tilde{p}[f_i]}{p[f_i]} \qquad i = 1,\dots,F,F+1$$

- Improved iterative scaling (IIS)

$$\sum_{m=1}^{M} \sum_{\{x|f_{\#}(x)=m\}} p(x) f_i(x) \beta_i^m = \tilde{p}[f_i] \qquad \beta_i = e^{\Delta\lambda_i}$$

Use Newton Method to solve the polynomial

课程章节

- 第一章 引言（1）
- 第二章 图模型的表示理论（2）
  - Semantics (DGM, UGM)
  - HMM, CRF
- 第三章 图模型的推理理论（6）
  - 精确推理：variable-elimination，cluster-tree，triangulate
  - 连续变量：Kalman
  - 采样近似：sampling
  - 变分近似：variational
- 第四章 图模型的学习理论（3）
  - 参数学习：maxlikelihoodEstimate，RFLearning, BayesEstimate
  - 结构学习：StructureLearning
- 第五章 一个综合例子（1）