

1. Intro

Hypergraph View: Computation graph = labeled acyclic hypergraph. Edges can have multiple sources/targets. **Complexity:** same time as f ; space higher (store intermediates) vec-vec: $O(d)$; mat-vec: $O(nm)$; mat-mat: $O(nml)$ **NLL** $\nabla = \mathbf{0}$: $\sum_{i=1}^n f(x_i, y_i) = \sum_{i=1}^n \mathbb{E}_{y|x_i, \theta}[f(x_i, y)]$ Observed features = Expected features **Hessian:** $H = \sum_i \text{Cov}_{y|x_i, \theta}[f(x_i, y)]$ (PSD!)

1. Backpropagation

Chain Rule: $\frac{d}{dx}[f(g(x))] = f'(g(x))g'(x)$ **Jacobian:** $f: \mathbb{R}^n \rightarrow \mathbb{R}^m, \frac{dy}{dx} = \left[\frac{dy_1}{dx_1}, \dots, \frac{dy_n}{dx_n} \right] \in \mathbb{R}^{m \times n}$ **Multivar:** $\frac{dy_i}{dx_j} = \sum_{k=1}^m \frac{dy_i}{dz_k} \frac{dz_k}{dx_j}$

Bauer Path Formula: $\frac{dy_i}{dx_j} = \sum_{p \in \mathcal{P}(j, i)} \prod_{(k, l) \in p} \frac{dz_l}{dx_k}$ $\mathcal{P}(j, i)$ =all paths $j \rightarrow i$; worst $O(m^n)$ **Computation Graph:** DAG w/ function nodes, edges=variable flow

Forward vs Reverse: **Forward:** expand $\frac{d}{dx}$ recursively, same flow as fwd **Reverse:** 2 passes—fwd compute vals, bwd compute grads **Complexity:** same time as f ; higher space (store intermediates)

Primitives: Sum: $\frac{d(a+b)}{da} = 1$; Prod: $\frac{d(ab)}{da} = b$

2. Log-Linear Models

Prob Basics: Bayes: $p(y|x) = \frac{p(x|y)p(y)}{\int p(x|y)p(y)dy}$ Marginal: $p(x) = \sum_y p(x, y)$ **Expectation:** $\mathbb{E}[f(x)] = \sum_x f(x)p(x)$

Log-Linear Model: $p(y|x, \theta) = \frac{\exp(\theta \cdot f(x, y))}{Z(\theta)}$, $Z(\theta) = \sum_{y' \in Y} \exp(\theta \cdot f(x, y'))$ $\log p(y|x, \theta) = \theta \cdot f(x, y) - \log Z$ (linear in log space!) **Discrete MLE:** $p(y|x) = \frac{\text{count}(x, y)}{\text{count}(x)}$ (sparse problem)

MLE $\nabla: \theta_{\text{MLE}} = \arg \min_{\theta} -\sum_{n=1}^N \log p(y_n|x_n, \theta) \frac{d\mathcal{L}}{d\theta} = -\sum_n f_k(x_n, y_n) + \sum_n \sum_{y'} p(y'|x_n; \theta) f_k(x_n, y')$ 观测特征计数 = 期望特征计数 → **Expectation Matching**

Softmax: $\text{softmax}(h, y, T) = \frac{\exp(h_y/T)}{\sum_{y'} \exp(h_{y'}/T)}$ $T \rightarrow 0$: argmax; $T \rightarrow \infty$: uniform $\log \text{softmax} = h_y - \log \sum_{y'} \exp(h_{y'})$ (logsumexp) $\frac{d \log \text{softmax}}{d\theta} = f(x, y) - \sum_i \text{softmax}(\theta^\top f, i) f(x, i)$

MLP Architecture: Problem: Log-linear needs linearly separable data **Solution:** Learn non-linear feature fn $\mathbf{h}_k = \sigma_k(\mathbf{W}_k^\top \mathbf{h}_{k-1})$, $\mathbf{h}_1 = \sigma_1(\mathbf{W}_1^\top e(x))$ Output: $\text{softmax}(\theta^\top \mathbf{h}_n)$

Sigmoid & Activations: $\sigma(x) = \frac{1}{1 + \exp(-x)}$, $\nabla \sigma = \sigma(1 - \sigma)$ $\tanh: \frac{1 - e^{-2x}}{1 + e^{-2x}}$, $\nabla = 1 - \tanh^2$. Sigmoid/tanh, vanishing gradient → use ReLU **Backprop(MLP):** $\frac{\partial \ell}{\partial \mathbf{W}_k} = \frac{\partial \ell}{\partial y} \frac{\partial y}{\partial \mathbf{h}_n} (\prod_{m=k+1}^M \sigma'_m \mathbf{W}_m) \sigma'_k \mathbf{h}_{k-1}$

Learning Pipeline: Embedding → Pooling (sum/mean/max) → NN → Softmax

Exp Family & MaxEnt: $p(x|\theta) = \frac{1}{Z(\theta)} h(x) \exp(\theta \cdot \varphi(x))$

Max Entropy: $H(p) = -\sum_x p(x) \log p(x)$ 选最大熵分布=最少假设=Laplace 原则

3. Language Models

Structured Prediction: Kleene V^* : infinite set of finite-length strings from V **Language Model:** weighted prefix tree, each sentence=unique path $p(y) = \frac{1}{Z} \prod_{t=1}^{|y|} \text{weight}_{y_t}$

Local Normalization: $Z = 1$ when children edges sum to 1 at each node **Consistency:** $p(\text{EOS}|y_{<t}, V^*) > \varepsilon > 0$ $p(|y| = \infty) \leq \lim_{t \rightarrow \infty} (1 - \varepsilon)^t = 0$ (tight)

N-gram Model: $p(y_t|y_{<t}) = p(y_t|y_{t-1}, \dots, y_{t-n+1}) = \frac{\exp(w_{y_t} h_t)}{\sum_{y' \in V} \exp(w_{y'} h_t)}$, $h_t \in \mathbb{R}^d$ **Bengio:** $h_t = f(e(\text{hist}))$, $e(\text{hist}) = [e(y_{t-1}); e(y_{t-2}); \dots]$

RNN: $h_t = f(h_{t-1}, e(y_{t-1}))$ (implicit infinite context)

Vanilla: $h_t = \sigma(W_h h_{t-1} + W_o e(y_{t-1}))$ **BPTT:** unroll through time, sum grads over timesteps

4. Word Embeddings

Encoding: One-hot: $v \in O(|V|)$, only word=1

Bag-of-words: pooled one-hot (sum/mean/max) **N-grams:** vectors huge—every combo needs slot

Skip-gram: Preprocess: word-context pairs ($k \times C$ many), window k $p(c|w) = \frac{1}{Z(w)} \exp(e_{\text{wrd}(w)} \cdot e_{\text{ctx}(c)})$, $O(2|V|k)$ params **Bilinear:** linear if all-but-one vars held constant **Similarity:** $\cos(u_i, u_j)$

5. CRF & POS Tagging

As Graph: Fully connected graph w/ POS-tag nodes per layer score($\langle D, N, V, \dots, w \rangle$) = $\theta f(t, w)$ Problem: $O(|\mathcal{T}|^N)$ paths in normalizer

CRF Model: $p(t|w) = \frac{\exp(\text{score}(t, w))}{\sum_{t' \in \mathcal{T}} \exp(\text{score}(t', w))}$

Decomposition: $\text{score}(t, w) = \sum_{n=1}^N \text{score}(\langle t_{n-1}, t_n \rangle, w, n)$ $p(t|w) \propto \prod_{n=1}^N \exp\{\text{score}(\langle t_{n-1}, t_n \rangle, w)\}$

Forward-Backward DP: $\forall t_n: \beta(w, t_N, N) \leftarrow 1$ for $n \leftarrow N-1, \dots, 0$: $\beta(w, t_n, n) \leftarrow \sum_{t_{n+1} \in \mathcal{T}} \exp(\text{score}) \times \beta(w, t_{n+1})$

Viterbi Decoding: $\beta(w, t_n) \leftarrow \max_{t_{n+1}} \exp(\text{score}) \times \beta(w, t_{n+1})$ **Structured CRF:** $\log p = \sum_i (\text{score}(t^{(i)}, w^{(i)}) - \max_{t'} \text{score}(t', w^{(i)}))$

DP 导引: Goal: $Z = \sum_t \exp \text{score}(t, w)$ **Step1:** 可加分解 $\text{score} = \sum_n \text{score}_n$ **Step2:** $Z = \sum_t \prod_n \exp \text{score}_n = \sum_{t_1} \exp \text{score}_1 \times \left(\sum_{t_2} \dots \right)$ (distrib!) $O(|\mathcal{T}|^N) \rightarrow O(N|\mathcal{T}|^2)$ 若依赖 3-gram: $O(N|\mathcal{T}|^3)$

Common Semirings:

Name	\mathbb{K}	\oplus	\otimes	0	1	用途
Real	$\mathbb{R}_{\geq 0}$	+	\times	0	1	Z partition fn
Viterbi	$\mathbb{R} \cup \{-\infty\}$	max	+	$-\infty$	0	最优 path/解码
Log	$\mathbb{R} \cup \{\pm\infty\}$	lse	+	$-\infty$	0	$\log Z$ 数值稳定
Boolean	$\{0, 1\}$	\vee	\wedge	0	1	可达性/存在性
Counting	\mathbb{N}	+	\times	0	1	路径数/歧义度
Tropical	$\mathbb{R} \cup \{\infty\}$	min	+	∞	0	最短路/编辑距离

Semiring Definition: $\langle \mathbb{K}, \oplus, \otimes, 0, 1 \rangle$ where:

1. $(\mathbb{K}, \oplus, 0)$: **comm monoid** (assoc+comm+identity)
2. $(\mathbb{K}, \otimes, 1)$: **monoid** (assoc+identity)
3. **Distrib:** $(x \oplus y) \otimes z = (x \otimes z) \oplus (y \otimes z)$
4. **Annihilator:** $0 \otimes x = x \otimes 0 = 0$

Semiring 意义: \oplus : 分治 (split points 合并, OR/MAX/+); **连接** (左右子树组合, AND/x/+); 0 : 吸收元, 消除 invalid; 1: 单位元, null 不破坏

Monoid 判定:

1. **Closure:** $a \otimes b \in \mathbb{K}$; 2. **Assoc:** $(a \otimes b) \otimes c = a \otimes (b \otimes c)$; 3. **Identity:** $\exists e: a \otimes e = e \otimes a = a$

Semiring 判定: 1. \oplus -monoid (comm): $a \oplus b = b \oplus a$; 2. \otimes -monoid; 3. Distributivity (左右皆需); 4. Annihilation: $0 \otimes x = 0$ 阴阱: $0 = 1$ 必失败!

Closed Semiring & Kleene*: $a^* = \bigoplus_{n=0}^{\infty} a^{\otimes n} = 1 \oplus a \otimes a^*$ Real $|a| < 1$: $a^* = \frac{1}{1-a}$ (geometric series) 用于全局 normalized LM

6. CFG Parsing

Constituents: Multi-word units as single unit **Tests:** Pronoun substitution, Clefting, Answer ellipsis Ambiguity: PP attachment, modifier scope

CFG Definition: $G = \langle \mathcal{N}, \mathcal{S}, \Sigma, \mathcal{R} \rangle$ Non-terminals, start symbol, terminals, production rules **CNF:** $N_1 \rightarrow N_2 N_3$ or $N \rightarrow a; O(4^N)$ trees

Weighted CFG: Global: $p(t) = \frac{1}{Z} \prod_{r \in t} \exp(\text{score}(r))$ $Z = \sum_{t' \in \mathcal{T}} \prod_{r'} \exp(\text{score}(r'))$ (可能 ∞ !) **Probabilistic:** local norm $\sum_k p(\alpha_k|N) = 1$

CKY algo: $O(N^3|R|)$, needs CNF for $n = 1, \dots, N$: for $X \rightarrow s_n \in \mathcal{R}$: chart[n, n+1, X] $\oplus \exp(\text{score}(X \rightarrow s_n))$ for span=2, ..., N: for $i = 1, \dots, N$: span: $k \leftarrow i + \text{span}$; for $j = i + 1, \dots, k - 1$: for $X \rightarrow Y Z \in \mathcal{R}$: chart[i, k, X] $\oplus \exp\{\text{score}\} \otimes \text{chart}[i, j, Y] \otimes \text{chart}[j, k, Z]$

Best parse: semiring (max, +)

7. Dependency Parsing

Dependency Tree: Directed spanning tree, root degree 1 **Projective:** no crossing arcs (~constituency w/ heads) **Non-projective:** crossing arcs (~discontinuous constituents) # spanning trees: $O((n-1)^{n-2})$

Edge-Factored Model: $p(t|w) = \frac{1}{Z} \prod_{(i \rightarrow j) \in t} \exp(\text{score}(i, j, w)) \exp(\text{score}(r, w))$

Edge factor assumption: score factors over edges

Matrix-Tree Theorem: $A_{ij} = \exp(\text{score}(i, j, w))$, $\rho_j = \exp(\text{score}(j, w))$ $Z = \det(L)$ where $L_{ij} = \begin{cases} \rho_j & i=1 \\ \sum_{i' \neq j} A_{i'j} & i=j \\ -A_{ij} & \text{else} \end{cases}$

Computing det in $O(n^3)$

MST Decoding: $\arg \max_{t \in \mathcal{T}} \sum_{(i \rightarrow j) \in t} \text{score}(i, j, w)$

Algo: max incoming edge, contract cycles (update weights) **Root Constraint:** for each root edge, compute removal cost; remove cheapest Runtime: $O(n^2)$

8. Semantic Parsing

Syntax vs Semantics: **Syntax:** structural org (parse tree)

Semantics: underlying meaning **Logical form:** quantifiers, vars, boolean, predicates

Lambda Calculus: **Abstraction:** M term, x var $\rightarrow \lambda x. M$ term **Application:** M, N terms $\rightarrow MN$ term

β -reduction: $(\lambda x. M)N = M[x := N]$ **β -infinity:** $F = \lambda x. (xx)x$, $FF = ((FF)F) = \dots$

Composition: $S_{VP} \rightarrow NP$ $VP, S_{sem} = VP.sem(NP.sem)$ **Compositionality:** meaning of whole = fn of parts

Combinatory Logic: $I: Ix = x$; $K: Kxy = x$; $S: Sxyz = (xz)(yz)$ S-K calculus = lambda calculus (via translator T)

Don't need I: $(SKK)x = x$

9. WFST & Lehmann

Transducer Def: $T = \langle Q, \Sigma, \Omega, \lambda, \rho, \delta \rangle$ Q : states; Σ : input vocab; Ω : output vocab $\lambda: Q \rightarrow \mathbb{R}$: initial weights; $\rho: Q \rightarrow \mathbb{R}$: final weights $\delta: Q \times (\Sigma \cup \varepsilon) \times (\Omega \cup \varepsilon) \times Q \rightarrow \mathbb{R}$ ε -transition: no input consumed / no output produced

FSA vs FST: **WFSA (单带):** read only, $\text{score}(\pi) = \sum_n \text{score}(\tau_n)$ **WFST (双带):** read input + write output simultaneously

Unambiguous: $|\Pi(x, y)| \leq 1$ (at most 1

path per pair)

Ambiguous: $|\Pi(x, y)| > 1 \rightarrow$ need semiring to aggregate

Path Score: $\text{score}(\pi) = \lambda(q_{\text{start}}) + \sum_{n=1}^{|\pi|} \text{score}(\tau_n) + \rho(q_{\text{end}})$ $p(y|x) = \frac{1}{Z} \sum_{\pi \in \Pi(x, y)} \exp(\text{score}(\pi))$ $Z = \sum_{y' \in \Omega} \sum_{\pi'} \exp(\text{score}(\pi'))$ (infinite!)

Matrix Multiplication View: $C = A \otimes B: C_{ij} = \bigoplus_k (A_{ik} \otimes B_{kj})$ **Tropical:** $C_{ij} = \min_k (A_{ik} + B_{kj})$ **Inside:** $C_{ij} = \sum_k (A_{ik} \times B_{kj})$ Naive $W^N: O(N^4)$ (N matrix mults $\times O(N^3)$ each)

Floyd-Warshall: Key: allow 中间 node k incrementally Runtime: $O(N^3)$ (reuses DP, not $O(N^4)$) $\text{dist}_k[i][j] = \min(\text{dist}_{k-1}[i][j], \text{dist}_{k-1}[i][k] + \text{dist}_{k-1}[k][j])$

Lehmann algo: **Generalized FW** 适用 any closed semiring: $\tilde{W}[i][j] \leftarrow W[i][j] \oplus (W[i][k] \otimes W[k][k]^* \otimes W[k][j])$ **Kleene Star** $W[k][k]^*$: self-loop at k any # times

- Tropical: $(W[k][k])^* = 0$ (positive cycle doesn't help)
- Inside: $(W[k][k])^* = \frac{1}{1 - W[k][k]}$ (geometric series)

Runtime: $O(|Q|^3)$

Kleene Star: $a^* = \bigoplus_{n=0}^{\infty} a^{\otimes n} = 1 \oplus a \otimes a^*$ **Closed semiring:** a^* always exists Real $|a| < 1$: $a^* = 1 + a + a^2 + \dots = \frac{1}{1 - a}$

Normalization Constant Z: $Z = \alpha^\top (\bigoplus_{\omega \in \Sigma^*} W^{(\omega)})^* \beta$

α : initial weight vec; β : final weight vec $W^{(\omega)}$: transition matrix for symbol ω **Why Lehmann?** Direct sum over infinite paths impossible Lehmann computes closure in $O(|Q|^3)$

Pathsum Formula: $Z(\mathcal{T}) = \bigoplus_{i, k \in Q} \lambda(q_i) \otimes R_{ik} \otimes \rho(q_k)$ **R:** all-pairs path weight matrix (via Lehmann)

Composition: $\mathcal{T}(x, y) = \bigoplus_{z \in \Omega^*} \mathcal{T}_1(x, z) \otimes \mathcal{T}_2(z, y)$

Transliteration: 3 transducers cascade \mathcal{T}_x (input) $\circ \mathcal{T}_\theta$ (model) $\circ \mathcal{T}_y$ (output)

10. Transformers & MT

Seq2Seq: $z = \text{encoder}(x)$, $y|x \sim \text{decoder}(z)$ $p(y|x) = \prod_{t=1}^T p(y_t|x, y_1, \dots, y_{t-1})$ Optimize log-likelihood

Attention: $\alpha^T V = \sum_i \alpha_i v_i^T$ (soft retrieval) $\alpha_i = \text{softmax}(\text{score}(q, k_i))$ $K = V = H^{(e)}$, $q_t = h_t^{(d)}$, $c = \alpha^T V$

Transformer Components: **Word Embed:** token→vector **Positional Embed:** encode word position (no recurrence!) **Residual Connections:** mitigate vanishing ∇ **Layer Norm:** normalize layer inputs **Self-attention:** Q, K, V from same sequence

Decoding: $y^* = \arg \max_{y \in \mathcal{Y}} \text{score}(x, y)$ W/o assumptions: $O(|\Sigma|^{n_{\text{max}}})$ paths **Beam Search:** keep k best at each step (greedy approx)

Sampling: sample from $p(y|x)$ each step **Eval:** BLEU (n-gram overlap), METEOR

11. Modeling Choices

Prob vs Non-Prob: **Prob:** leverage prob theory, needs assumptions CRF, RNN, N-gram models **Non-Prob:** interpretable, uncertainty unclear Perceptron, SVM, CFG rules

Disc vs Generative: **Discriminative:** model boundary $p(y|x)$ **Generative:** model own dist $p(x, y)$

Local vs Global Norm: **Local:** efficient train, biased predictions **Global:** needs Z , unbiased Independence assumptions control complexity

Loss & Regularization: **LogLoss:** $\ell(y, y') = \log(1 + e^{-y'y})$ **Exp-Loss:** $\ell(y, y') = e^{-y'y}$ **L1/L2:** weight penalties (Laplace/Gaussian prior)

Evaluation Metrics: **Prec:** $P_{\text{true}}/P_{\text{all}}$; **Recall:** $P_{\text{true}}/(P_{\text{true}} + N_{\text{false}})$ **Acc:** $(P_{\text{true}} + N_{\text{true}})/N$ **F-score:** $((1 + \beta^2)(\text{prec} \cdot \text{recall})) / (\beta^2 \text{prec} + \text{recall})$

Statistical Tests: $p = 2 \min(P(T \geq t|H_0), P(T \leq t|H_0))$; Rej if $p < \alpha$ **Power:** $P(\text{reject } H_0|H_1)$ **Multiple tests:** $P(|\text{FalseRej}| > 0) = 1 - (1 - \alpha)^K$ **Bonferroni:** $\alpha^* = \alpha/K$

Permutation Test: p^* from original data; permute labels k times p-value= $(|\{i : p_i \leq p^*\}| + 1)/(k + 1)$

McNemar's Test: $\chi^2 = \frac{(b-c)^2}{b+c} \sim \chi^2_1$ for $b, c \geq 25$ $H_0 : p_b = p_c$; $b=C1$ wrong/ $C2$ right

5x2cv Test: $\bar{p} = (p^{(1)} + p^{(2)})/2$ $s^2 = (p^{(1)} - \bar{p})^2 + (p^{(2)} - \bar{p})^2$ $t = (p_1^{(1)})/\sqrt{1/5 \sum s_i^2} \sim t^5$

12. Bias & Fairness

Bias Sources: **Labeling:** reproduce annotator bias **Sample selection:** training fits certain profile **Task definition:** excludes certain groups **Imbalanced test:** loss ignores minorities

Ethical Frameworks: **Consequentialism:** best consequence **Utilitarianism:** hedonistic/preference/welfare **Deontologism:** rules must be kept **Social Contract:** natural equality **Anti-subordination:** positive discrimination for equality

Quick Ref **Chain:** $\frac{d}{dx}[f(g(x))] = f'(g)g'(x)$; **Bauer:** sum over all paths **Softmax:** $\exp(h_y)/\sum \exp(h_{y'})$; $T \rightarrow 0 = \text{argmax}$; $T \rightarrow \infty = \text{uniform}$ **Log-Linear:** $p(y|x) = \exp(\theta \cdot f)/Z$; MLE matches expected features **CRF:** decompose score \rightarrow DP; semiring unifies algos **Viterbi:** max instead of sum; decoding= arg max score **CKY:** $O(N^3|R|)$; needs CNF; semiring for best/count/prob **Dep Parse:** Matrix-Tree for Z in $O(n^3)$; MST+contract cycles **Attention:** $\alpha = \text{softmax}(QK^T)$; $c = \alpha V$ (soft lookup) **WFST:** Kleene* for infinite sums; $(I - W)^{-1}$ **Lambda:** β -reduction $(\lambda x.M)N = M[x := N]$ **Local vs Global:** bias vs intractability tradeoff **Semirings:** Boolean/Viterbi/Inside/Tropical/Counting **Stats:** Bonferroni α/K , McNemar $(b - c)^2/(b + c)$

Abbrev **BOS/EOS:** Begin/End of Sentence; **CCG:** Combinatory Categorical Grammar; **CFG:** Context-Free Grammar; **CKY:** Cocke-Kasami-Younger; **CNF:** Chomsky Normal Form; **CRF:** Conditional Random Field; **DP:** Dynamic Programming; **LLM:** Log-Linear Model; **MLE:** Max Likelihood Est; **MST:** Min Spanning Tree; **NLP:** Natural Language Processing; **POS:** Part-of-Speech; **RNN:** Recurrent Neural Network; **WFST:** Weighted Finite State Transducer;