0. 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: $O(nm\ell)$

NLL $\nabla = 0$: $\sum_{i=1}^n \boldsymbol{f}(x_i,y_i) = \sum_{i=1}^n \mathbb{E}_{y|x_i,\theta}[\boldsymbol{f}(x_i,y)]$ Observed features = Expected features **Hessian**: $\boldsymbol{H} = \sum_i \mathrm{Cov}_{y|x_i,\theta}[\boldsymbol{f}(x_i,y)]$ (PSD!)

1. Backpropagation

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

Bauer Path Formula: $\frac{dy_i}{dx_j} = \sum_{p\in\mathcal{P}(j,i)} \prod_{(k,l)\in p} \frac{dz_l}{dz_k}$ $\mathcal{P}(j,i)$=all paths $j\to i$; 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 问题)

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

Softmax: $\text{softmax}(h,y,T) = \frac{\exp(h_y/T)}{\sum_{y'}\exp(h_{y'}/T)}$ $T\to 0$: argmax; $T\to\infty$: 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 $\boldsymbol{h}_k = \sigma_k(\boldsymbol{W}_k^\top \boldsymbol{h}_{k-1})$, $\boldsymbol{h}_1 = \sigma_1(\boldsymbol{W}_1^\top \boldsymbol{e}(x))$ Output: $\text{softmax}(\boldsymbol{\theta}^\top\boldsymbol{h}_n)$

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)**: $\frac{\partial\ell}{\partial\boldsymbol{W}_k} = \frac{\partial\ell}{\partial y}\frac{\partial y}{\partial\boldsymbol{h}_n}\left(\prod_{m=k+1}^n \sigma_m'\boldsymbol{W}_m\right)\sigma_k'\boldsymbol{h}_{k-1}$

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 $p(y) = \frac{1}{Z}\prod_{t=1}^{|y|}\text{weight}_{y_{\le t}}$

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

N-gram Model: $p(y_t|y_{<t}) = p(y_t|y_{t-1},\dots,y_{t-n+1}) = \frac{\exp(w_{y_t}\cdot h_t)}{\sum_{y'\in V}\exp(w_{y'}\cdot h_t)}$, $h_t\in\mathbb{R}^d$ **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**: $h_t = \sigma(W_1 h_{t-1} + W_2 e(y_{t-1}))$ **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 $p(c|w) = \frac{1}{Z(w)}\exp\left(e_{\text{wrd}(w)}\cdot e_{\text{ctx}(c)}\right)$, $O(2|V|k)$ 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 $\text{score}(\langle D,N,V,\dots\rangle, w) = \theta f(t,w)$ **Problem**: $O(|\mathcal{T}|^N)$ paths in normalizer

CRF Model: $p(t|w) = \frac{\exp(\text{score}(t,w))}{\sum_{t'\in\mathcal{T}^N}\exp(\text{score}(t',w))}$ **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**: $\log p = \sum_i\left(\text{score}(t^{(i)},w^{(i)}) - \max_{t'}\text{score}(t',w^{(i)})\right)$

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

Semiring 意义: ⊕: **分治** (split points 合并, OR/MAX/+) ⊗: **连接** (左右子树组合, AND/×/+) **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**: $\exists\boldsymbol{e}: a\otimes\boldsymbol{e} = \boldsymbol{e}\otimes a = a$

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

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

DP 推导: Goal: $Z = \sum_{\boldsymbol{t}}\exp\text{score}(\boldsymbol{t},\boldsymbol{w})$ **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!) $O(|\mathcal{T}|^N)\to O(N|\mathcal{T}|^2)$ **若依赖 3-gram**: $O(N|\mathcal{T}|^3)$

Common Semirings:

Name	$\mathbb{K}$	⊕	⊗	0	1	用途
Real	$\mathbb{R}_{\ge0}$	+	×	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	$\{0,1\}$	∨	∧	0	1	可达性/存在性
Counting	$\mathbb{N}$	+	×	0	1	路径数/歧义度
Tropical	$\mathbb{R}\cup\{\infty\}$	min	+	∞	0	最短路/编辑距离

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**: $N_1\to N_2N_3$ or $N\to a$; $O(4^N)$ 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 Algorithm: $O(N^3|R|)$, needs CNF for $n=1,\dots,N$: for $X\to s_n\in\mathcal{R}$: chart$[n,n+1,X]\oplus\exp(\text{score}(X\to s_n))$ for span$=2,\dots,N$: for $i=1,\dots,N-$span: $k\leftarrow i+$span; for $j=i+1,\dots,k-1$: for $X\to YZ\in\mathcal{R}$: chart$[i,k,X]\oplus\exp\{\text{score}\}\otimes$chart$[i,j,Y]\otimes$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: $p(t|w) = \frac{1}{Z}\prod_{(i\to j)\in t}\exp(\text{score}(i,j,w))\exp(\text{score}(r,w))$ **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 $L_{ij} = \begin{cases}\rho_j & i=1\\ \sum_{i'\ne j}A_{i'j} & i=j\\ -A_{ij} & \text{else}\end{cases}$ Computing det in $O(n^3)$

MST Decoding: $\arg\max_{t\in\mathcal{T}}\sum_{(i\to j)\in t}\text{score}(i,j,w)$ **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 → $\lambda x.M$ term **Application**: M,N terms → 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} →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. WFSTs

Transducer: $T = \langle\mathcal{Q},\Sigma,\Omega,\lambda,\rho,\delta\rangle$ States, input vocab, output vocab, initial/final scores, transitions Goal: $p(y|x)$, $x\in\Sigma^*, y\in\Omega^*$

Scoring: $\text{score}(\pi) = \lambda(q_{\text{start}}) + \sum_n\delta(q_n) + \rho(q_{\text{end}})$ $p(y|x) = \frac{1}{Z}\sum_{\pi\in\Pi(x,y)}\exp(\text{score}(\pi))$ $Z = \sum_{y'\in\Omega^*}\sum_{\pi'}\exp(\text{score})$ (infinite—loops!)

Floyd-Warshall + Semiring: $\forall i,j,k$: dist$[i][j]\leftarrow$ dist$[i][j]\oplus($dist$[i][k]\otimes$dist$[k][j])$ **Matrix mult**: sum← $\bar{0}$; sum←sum$\oplus(A[n][m]\otimes B[m][p])$

Kleene Star: $W^* = \bigoplus_{k=0}^\infty W^k = I + WW^* \Leftrightarrow W^* = (I-W)^{-1}$ **Warshall-Floyd-Kleene**: dist$[i][j]\leftarrow$ dist$[i][j]\oplus($dist$[i][k]\otimes$dist$[k][k]^*\otimes$dist$[k][j])$

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 ∇s **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: $O(|\Sigma|^{n_{\max}})$ 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**: $\ell(y,y') = \log\left(1+e^{-y\cdot y'}\right)$ **Exp-Loss**: $\ell(y,y') = e^{-y\cdot y'}$ **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**: $\frac{(1+\beta^2)(\text{prec}\cdot\text{recall})}{\beta^2\text{prec}+\text{recall}}$

Statistical Tests: $p = 2\min(P(T\ge t|H_0), P(T\le t|H_0))$; 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$=(|\{i: p_i\le p^*\}|+1)/(k+1)$

McNemar's Test: $\chi^2 = \frac{(b-c)^2}{b+c}\sim\chi_1^2$ for $b,c\ge 25$ H_0: $p_b = p_c$; b=C1 wrong/C2 right

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

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 **Utilitarism**: hedonistic/preference/welfare **Deontology**: 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**: $\exp(h_y)/\sum\exp(h_{y'})$; $T\to 0$=argmax; $T\to\infty$=uniform **Log-Linear**: $p(y|x) = \exp(\theta\cdot f)/Z$; MLE matches expected features **CRF**: decompose score → 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 Categorial 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;