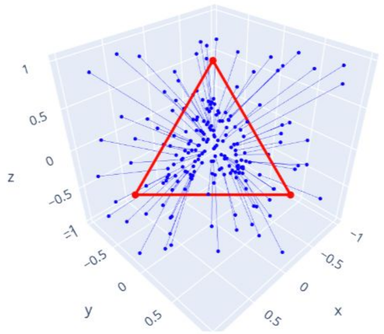
Notation

- Input **logit** vector in K dimensions
- Normalize by exp **log-partition** *A*

$$\text{softmax}(\ell) = \frac{\exp \ell_i}{\exp A(\ell)}$$

- Exponential family identity

$$A'_{\ell_i}(\ell) = \text{softmax}(\ell)_i$$



Conditional Random Fields

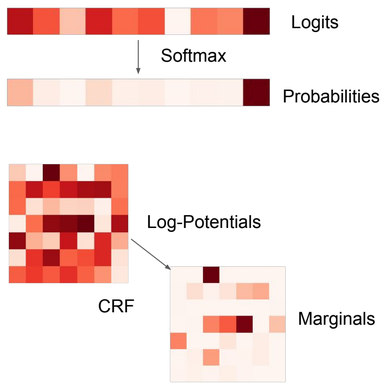
Softmax for combinatorial distributions

- Softmax: Logits -> Partition -> Probabilities

$$A'_{\ell_i}(\ell) = \text{softmax}(\ell)_i$$

- CRF: Log-Potentials -> Partition -> Marginals

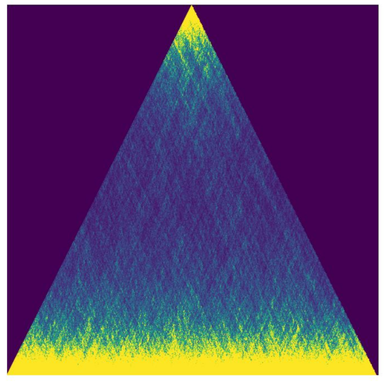
$$A'_{\ell_i}(\ell) = p(x_i = 1)$$



Torch-Struct

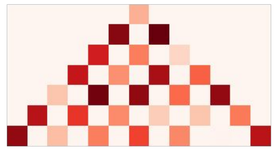
https://github.com/harvardnlp/pytorch-struct

- Library for structured softmax / CRF.
- Goal: CRF *as easy as* Softmax
- Challenge: Inference as *fast as* Softmax

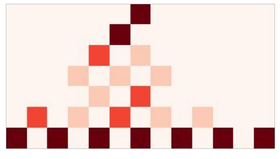


User Interface: Tree CRF

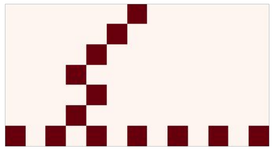
log_potentials



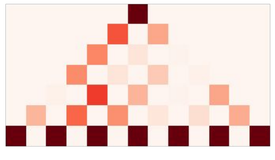
TreeCRF(log_potentials).sample((5,))



TreeCRF(log_potentials).argmax



TreeCRF(log_potentials).marginals



Methodology

- Algorithms -> Vectorized Log-Partition

$$A(\ell)$$

- Marginals -> Computed with autodiff

$$A'_{\ell_i}(\ell)$$

- Semirings -> Generic algorithms

$$A^*(\ell)$$

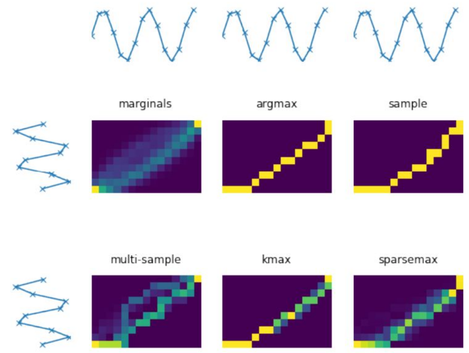
```
def tree_log_partition(log_potentials):
    ...
    # Tree log partition implementation
    for w in range(1, N):
        left = slice(None, N - w)
        right = slice(w, None)
        Y = beta[A][left, :w]
        Z = beta[B][right, N - w :]
        score = reduced_scores.diagonal(w, L_DIM, R_DIM)
        new = semiring.times(semiring.dot(Y, Z), score)
        beta[A][left, w] = new
        beta[B][right, N - w - 1] = new

    ...
    return log_partition
```

Library of Models

Algorithms $A(\ell)$

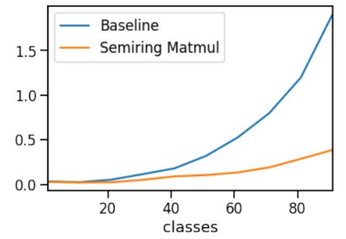
Name	Structure ($\mathcal{Z}$)
Linear-Chain	Labeled Chain
Factorial-HMM	Labeled Chains
Alignment	Alignment
Semi-Markov	Seg. Labels
Context-Free	Labeled Tree
Simple CKY	Labeled Tree
Dependency	Proj. Tree
Dependency (NP)	Non-Proj. Tree
Auto-Regressive	Sequence

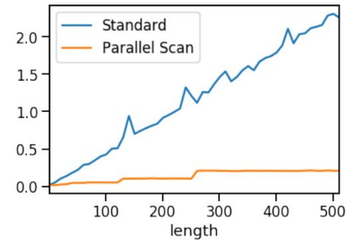
Semirings $A^*(\ell)$



Optimizations

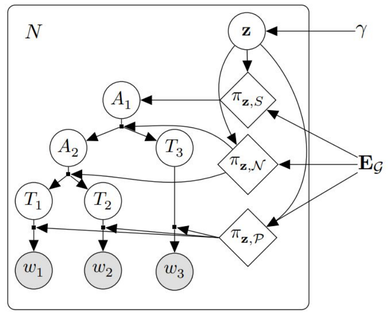
- Custom semiring CUDA kernels
- Parallel scan forward-backward
- Vectorization of inside-outside
- Recent: JAX/XLA compiled versions





Applications

Compound PCFG



HMM-NNLM