Unifying Sum-Product Networks and Submodular Fields

(Supplementary Material)

Abram L. Friesen [1] **Pedro Domingos** [1]

1. Additional model details

1.1. SSPN to SPN conversion

An SSPN defines an SPN containing a sum node for each possible region of each nonterminal, a product node for each segmentation of each production of each possible region of each nonterminal, and a leaf function on the pixels of the image for each possible region of the image for each terminal symbol. The children of the sum node s for nonterminal X_s with region $\mathcal{R}_s$ are all product nodes r with a production $v_r : X_s \rightarrow Y_1 \dots Y_k$ and region $\mathcal{R}_{v_r} = \mathcal{R}_s$. Each product node corresponds to a labeling $\mathbf{y}^{v_r}$ of $\mathcal{R}_{v_r}$ and the edge to its parent sum node has weight $\exp(-w_v - E(\mathbf{y}^{v_r}, \mathcal{R}_{v_r}))$. The children of product node r are the sum or leaf nodes with matching regions that correspond to the constituent nonterminals or terminals of v_r, respectively. Note that this underlying SPN is decomposable, but not smooth. However, (Friesen & Domingos, 2016) showed that smoothness was not a necessary condition for tractable inference and that no corrective factor is necessary when operating in the min-sum semiring, which is what is used for finding the (approximate) optimal parse of an SSPN.

A key benefit of SSPNs in comparison to previous grammar-based approaches is that regions can have arbitrary shapes and are not restricted to a small class of shapes such as rectangles (Poon & Domingos, 2011; Zhao & Zhu, 2011). This flexibility is important when parsing images, as real-world objects and abstractions can take any shape, but it comes with a combinatorial explosion of possible parses. However, by exploiting submodularity, we are able to develop an efficient inference algorithm for SSPNs, allowing us to efficiently parse images into a hierarchy of arbitrarily-shaped regions and objects, yielding a very expressive model class. This efficiency is despite the size of the underlying SSPN, which is in general far too large to explicitly instantiate.

[1]Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, WA USA. Correspondence to: Abram Friesen <afriesen@cs.washington.edu>, Pedro Domingos <pedrod@cs.washington.edu>.

Presented at the ICML 2017 Workshop on Principled Approaches to Deep Learning, Sydney, Australia, 2017.

2. Proofs

Proposition 1. *The energy $E(v, t_1, t_2, \mathbf{y}^v)$ of the fusion of parse trees t_1, t_2 over region $\mathcal{R}$ with head symbols Y_1, Y_2 for a production $v : X \rightarrow Y_1 Y_2$ is submodular.*

Proof. $E(v, t_1, t_2)$ is submodular as long as $2 \cdot \theta^v_{pq}(Y_1, Y_2) \geq \theta^{t_1}_{pq} + \theta^{t_2}_{pq}$, where $\theta^t_{pq} = \sum_{u \in t} \theta^u_{pq}(y^u_p, y^u_q) \cdot [(p, q) \in \mathcal{E}_u]$. Let every submodular MRF energy $E^u(\mathbf{y}^u, \mathcal{R})$ be in *normal* form such that $\theta^u_{pq}(y, y) = 0$ and $\theta^u_{pq}(y^u_p, y^u_q) \geq 0$ for all labels $y, y^u_p, y^u_q \in \mathcal{Y}_u$, where any submodular energy can be reparameterized into normal form in time linear in the region size (Kolmogorov & Rother, 2007). Then, since θ^t_{pq} can contain at most one production $c \in t$ such that $y^c_p \neq y^c_q$, it follows that θ^t_{pq} contains at most one non-zero term, in which case $\theta^t_{pq} = \theta^c_{pq}$. Finally, since $\theta^v_{pq}(y^v_p, y^v_q) \geq \theta^c_{pq}(y^c_p, y^c_q)$ for c any possible descendant of v and for all labelings, then $\theta^v_{pq}(Y_1, Y_2) \geq \theta^{t_i}_{pq}$ for $i \in \{1, 2\}$ and the claim follows. □

The following result shows how INFERSSPN can improve a parse tree while ensuring that the energy of that parse tree never gets worse.

Lemma 1. *Given a labeling $\mathbf{y}^v$ which fuses parse trees t_1, t_2 into t with root production v, energy $E(t, \mathcal{R}) = E(v, t_1, t_2, \mathbf{y}^v)$, and subtree regions $\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset$ defined by $\mathbf{y}^v$, then any improvement Δ in $E(t_1, \mathcal{R}_1)$ also improves $E(t, \mathcal{R})$ by at least Δ, regardless of any change in $E(t_1, \mathcal{R} \backslash \mathcal{R}_1)$.*

Proof. Since the optimal fusion can be found exactly, and the energy of the current labeling $\mathbf{y}^v$ has improved by Δ, the optimal fusion will have improved by at least Δ. □

Proposition 2. *Let $c(n)$ be the time complexity of computing a graph cut on n pixels and $|G|$ be the size of the gram-*

mar defining the SSPN, then each iteration of INFERSSPN *takes time* $O(|G|c(n)n)$.

Proof. Let k be the number of productions per nonterminal symbol and N be the nonterminals. The three main loops of the algorithm have complexity $|N|$, n (because there can be at most n regions and the regions are disjoint), and k, respectively. For line 8, the choice of parses for productions in $\hat{t}$ takes constant time, and the rest can be chosen arbitrarily. For lines 9-10, the fusion of a region $\mathcal{R}$ has complexity $O(|\mathcal{R}| + c(|\mathcal{R}|) = O(c(|\mathcal{R}|))$, so the worst-case complexity of the inner loop is when $\mathcal{R}$ is empty or the full image, giving complexity $O(c(n))$. Thus, the total complexity of each iteration of INFERSSPN is $O(|N|k \cdot c(n) \cdot n) = O(|G|c(n)n)$. □

Theorem 1. *Given a parse (tree)* $\hat{t}$ *of* S *over the entire image with energy* $E(\hat{t})$*, each iteration of* INFERSSPN *constructs a parse (tree)* t *of* S *over the entire image with energy* $E(t) \leq E(\hat{t})$*, and since the minimum energy of an image parse is finite,* INFERSSPN *will always converge.*

Proof. We will prove by induction that for all nodes $n \in \hat{t}$ with corresponding production $v : X \to YZ$, region $\mathcal{R}_X$, subtree $\hat{t}_{\mathcal{R}_X}$ over region $\mathcal{R}_X$, and child subtrees $\hat{t}_{\mathcal{R}_Y}, \hat{t}_{\mathcal{R}_Z}$ over regions $\mathcal{R}_Y, \mathcal{R}_Z$, that $E(t_{\mathcal{R}_X}) \leq E(\hat{t}_{\mathcal{R}_X})$ after one iteration. Since the start symbol S has only one region containing the entire image, this proves the claim.

Base case. Let $\hat{t}_{\mathcal{R}_X}$ be a subtree with region $\mathcal{R}_X$ and production $v : X \to Y$ containing only a single terminal child and let $\{u_i = X \to Y_i\}$ be the set of productions of X (where such a $\hat{t}_{\mathcal{R}_X}$ must exist because the grammar is non-recursive and terminates). By definition, $t_v = \hat{t}_{\mathcal{R}_X}$, where t_v is the new parse of $\mathcal{R}_X$ as v, because terminal parses do not change for the same region. Then, since $t_{\mathcal{R}_X} = \arg\min_{u_i} E(t_{u_i})$ and $v \in \{u_i\}$, it immediately follows that $E(t_{\mathcal{R}_X}) \leq E(\hat{t}_{\mathcal{R}_X})$ and the claim holds.

Induction step. Let $n \in \hat{t}$ be a node in $\hat{t}$ with corresponding production $v : X \to YZ$, region $\mathcal{R}_X$, subtree $\hat{t}_{\mathcal{R}_X}$ over region $\mathcal{R}_X$, and child subtrees $\hat{t}_{\mathcal{R}_Y}, \hat{t}_{\mathcal{R}_Z}$ over regions $\mathcal{R}_Y, \mathcal{R}_Z$, such that $\mathcal{R}_Y \cup \mathcal{R}_Z = \mathcal{R}_X$ and $\mathcal{R}_Y \cap \mathcal{R}_Z = \emptyset$, and suppose that $E(t_{\mathcal{R}_Y}) \leq E(\hat{t}_{\mathcal{R}_Y})$ and $E(t_{\mathcal{R}_Z}) \leq E(\hat{t}_{\mathcal{R}_Z})$. From Lemma 1, it follows that the parse t_v computed from fusing $t_{\mathcal{R}_Y}$ and $t_{\mathcal{R}_Z}$ in $\mathcal{R}_X$ as v has energy $E(t_v) \leq E(\hat{t}_{\mathcal{R}_X})$ (since the fusion can always choose the same labeling as in $\hat{t}_{\mathcal{R}_Y}$). Then, since $t_{\mathcal{R}_X} = \arg\min_{u \in \{u_X : \text{head}(u_X) = X\}} E(t_u)$, where $\{u_X\}$ are the productions of X, we have that $E(t_{\mathcal{R}_X}) \leq E(t_v)$ and thus $E(t_{\mathcal{R}_X}) \leq E(\hat{t}_{\mathcal{R}_X})$ and the claim follows. □

3. Additional experimental details and results

3.1. Additional figures

Figures 1, 2, and 3 show the full matrix of the performance of INFERSSPN, α-expansion, and BP for each measure (minimum energy found, parsing time taken, and mean average pixel accuracy) of the three scenarios (varying the strength of boundary terms, increasing the grammar height, and increasing the number of productions for each nonterminal) described in the main paper.

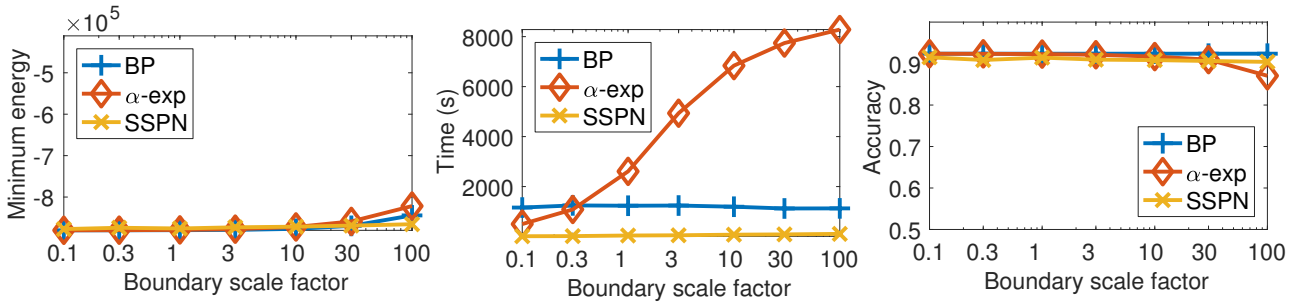
3.2. MRF segmentation details

As discussed in the main paper, the energy of each segmentation of a region for a given production is defined by a MRF $E(\mathbf{y}^v, \mathcal{R}_v) = \sum_{p \in \mathcal{R}_v} \theta_p^v(y_p^v; \mathbf{w}) + \sum_{(p,q) \in \mathcal{E}_v} \theta_{pq}^v(y_p^v, y_q^v; \mathbf{w})$. The unary and pairwise terms in E can be defined arbitrarily, as long as the resulting energy is submodular. In our experiments, we define the unary terms for terminals $T \in \Sigma$ as a linear function of the image features $\theta_p^v(y_p^v = T; \mathbf{w}) = \mathbf{w}_T^\top \phi_p^U$, where ϕ_p^U is a feature vector representing the local appearance of pixel p. Unary terms for nonterminals $X \in N$ can be defined as $\theta_p^v(y_p^v = X; \mathbf{w}) = w_{pX}^v$, where w_{pX}^v is a (learnable) parameter that specifies how likely this pixel is to be labeled as X. This allows each production to encode the regions of the image associated with each of its constituents.

The pairwise terms are also quite flexible, but in our experiments we use the standard contrast-dependent pairwise boundary potential (e.g., Shotton et al. (2006)) defined for each production v and each pair of adjacent pixels p, q as $\theta_{pq}^v(y_p^v, y_q^v; \mathbf{w}) = w_v^{\text{BF}} \exp(-\beta^{-1} ||\phi_p^B - \phi_q^B||^2) \cdot [y_p^v \neq y_q^v]$, where β is half the average image contrast between all adjacent pixels in an image, w_v^{BF} is the boundary factor that controls the relative cost of this term for each production, ϕ_p^B is the pairwise per-pixel feature vector, and $[\cdot]$ is the indicator function, which has value 1 when its argument is true and is 0 otherwise.

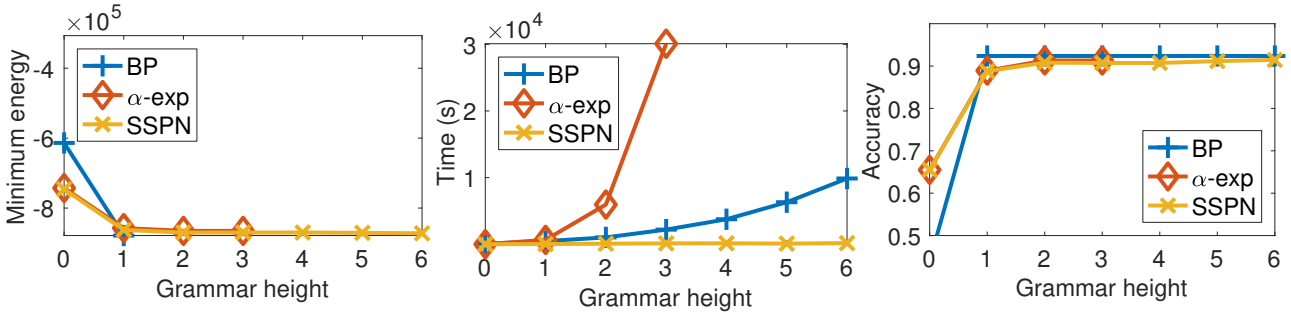
3.3. α-expansion and 3-D MRF details

We compared INFERSSPN to running α-expansion on a flat pairwise MRF and to max-product belief propagation over a multi-level (3-D) pairwise grid MRF. Each label of the flat MRF corresponds to a possible path in the grammar from the start symbol to a production to one of its constituent symbols, etc, until reaching a terminal. In general, the number of such paths is exponential in the height of the grammar. The unary terms are the sum of unary terms along the path and the pairwise term for a pair of labels is the pairwise term of the first production at which their constituents differ. For any two labels with paths that choose a different production of the same symbol (and have the same path from the start symbol) we assign infinite cost to



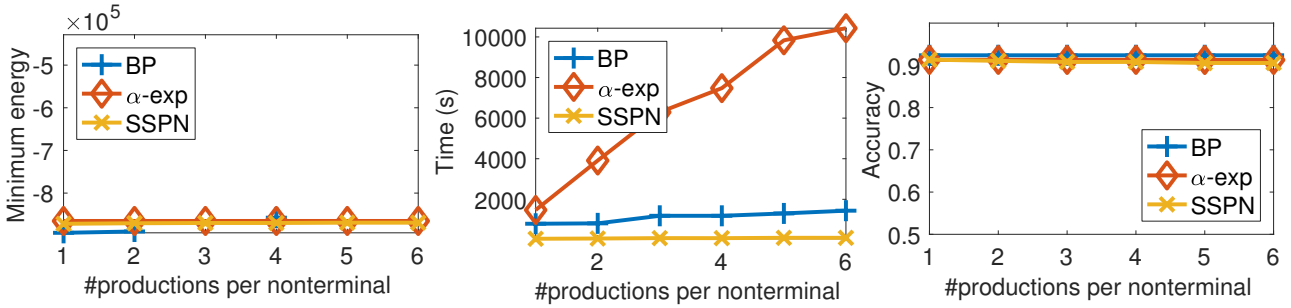
(a) (b) (c)

Figure 1. The (a) best energy, (b) total running time, and (c) resulting semantic segmentation accuracy (mean average pixel accuracy) for belief propagation, α-expansion, and INFERSSPN when varying boundary strength. Each data point is the average value over (the same) 10 images.



(a) (b) (c)

Figure 2. The (a) best energy, (b) total running time, and (c) resulting semantic segmentation accuracy (mean average pixel accuracy) for belief propagation, α-expansion, and INFERSSPN when varying grammar height. Each data point is the average value over (the same) 10 images. Missing data points for α-expansion indicate that it ran out of memory. Missing data points for BP indicate that it returned infinite energy (left). Low accuracies for grammar height 0 are a result of the grammar being insufficiently expressive.



(a) (b) (c)

Figure 3. The (a) best energy, (b) total running time, and (c) resulting semantic segmentation accuracy (mean average pixel accuracy) for belief propagation, α-expansion, and INFERSSPN when varying grammar height. Each data point is the average value over (the same) 10 images. Missing data points for BP indicate that it returned infinite energy (left).

enforce the restriction that an object can only have a single production of it into constituents. Note that after convergence α-expansion is guaranteed to be within a constant factor of the global minimum energy (Boykov et al., 2001) and thus serves as a good surrogate for the true global minimum, which is intractable to compute. The multi-layer MRF is constructed similarly. The number of levels in the MRF is equal to the height of the DAG corresponding to the grammar used. The labels at a particular level of the MRF are all (production, constituent) pairs that can occur at this height in the grammar. The pairwise term between the same pixel in two levels is 0 when the parent label's constituent equals the child label's production head, and ∞ otherwise. Pairwise terms within a layer are defined as in the flat MRF with infinite cost for incompatible labels (i.e., two neighboring pixels parsed as different productions of the same symbol), unless two copies of that nonterminal could be produced at that level by the grammar.

3.4. Details on inference evaluation experiments

We compared the three inference algorithms by varying three different parameters: boundary strength (strength of pairwise terms), grammar height, and number of productions per nonterminal. Each grammar contained a start symbol, multiple layers of nonterminals, and a final layer of nonterminals in one-to-one correspondence with the eight terminal symbols, each of which had a single production that produces a region of pixels. The start symbol had one production for each pair of symbols in the layer below it, and the last nonterminal layer (ignoring the nonterminals for the labels) had productions for each pair of labels, distributed uniformly over this last nonterminal layer. The productions between intermediate layers are generated randomly.

All experiments were run on the same computer running a dual 20-core 2.2 GHz Intel Xeon E5-2698 and 512 GB of RAM. Each algorithm was limited to a single thread.

Boundary strength. Increasing the boundary strength of an MRF makes inference more challenging, as individual pixel labels cannot be easily flipped without large side effects. To test this, we constructed a grammar as above with 2 layers of nonterminals (not including the start symbol), each containing 3 nonterminal symbols with 4 binary productions to the next layer. We used a single weight w_{BF} to parameterize all pairwise (boundary) terms in the MRF of every production. Figure 1 plots the mean average pixel accuracy of the parses returned by each algorithm vs. w_{BF} (the x-axis is log-scale). INFERSSPN returns parses with almost identical accuracy (and energy) to α-expansion. BP also returns comparable accuracies, but almost always returns invalid parses with infinite energy (if it converges at all) that contain multiple productions of the same object or a production of a symbol Y even though the pixel is labeled as symbol X.

Grammar height. In general, the number of paths in the grammar is exponential in its height, so the height of the grammar controls the complexity of inference and thus the difficulty of parsing images. For this experiment, we set w_{BF} to 20 and constructed a grammar with four nonterminals per layer, each with three binary productions to the next layer. Figure 2 shows the effect of grammar height on total inference time (to convergence or a maximum number of iterations, whichever first occurred). As expected from Proposition 1, the time taken for INFERSSPN scales linearly with the height of the grammar, which is within a constant factor of the size of the grammar when all other parameters are fixed. Similarly, inference time for both α-expansion and BP scaled exponentially with the height of the grammar because the number of labels for both increases combinatorially. Again, the energies and corresponding accuracies achieved by INFERSSPN were nearly identical to those of α-expansion (see Figure 2, below).

Productions per nonterminal. The number of paths in the grammar is also directly affected by the number of productions per symbol. For this experiment, we set w_{BF} to 20 and constructed a grammar with 2 layers of nonterminals, each with 4 nonterminal symbols. Figure 3 shows the effect of increasing the number of productions per nonterminal, which again demonstrates that INFERSSPN is far more efficient than either α-expansion or BP as the complexity of the grammar increases, while still finding comparable solutions (see Figure 3, below).

3.5. Details on model evaluation and grammar induction

To induce a grammar for a particular image, we first constructed 4 segmentations of the image at increasing levels of granularity using the method of Isola et al. (2014) and then intersecting these regions with the regions from the true labels. We then did the same for 4 other images chosen uniformly at random. The segments from these 5 images define the symbols of the induced grammar and the regions of the segments determine the regions of the symbols in the per-pixel MRF weights. We next created productions between overlapping segments at each neighboring levels of segmentation granularity within the same image. We then generated 3 productions for each symbol by randomly selecting 4 regions in the next level of segmentation granularity across all images that overlapped with the head symbol's region. Finally, we set the produced terminals of the finest-granularity regions to be those labels that were expressed anywhere in their region, as these form the bottom of the grammar. For the MRF weights, we set w_{BF} to 5 for all productions and all edges, but used the contrast-dependent pairwise boundary potential defined above to control the strength of the pairwise terms. The produced DeepLab features from the layer preceding the softmax (in the DeepLab architecture) were used as the per-pixel unary costs for productions into the corresponding terminal symbol. Production costs were all set to 0, so the energy is entirely determined by the per-pixel unary costs, the pairwise costs, and the structure of the grammar.

References

Boykov, Yuri, Veksler, Olga, and Zabih, Ramin. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

Friesen, Abram L. and Domingos, Pedro. The sum-product theorem: A foundation for learning tractable models. In *Proceedings of the 33rd International Conference on Machine Learning (ICML 2016)*, volume 48, 2016.

Isola, Phillip, Zoran, Daniel, Krishnan, Dilip, and Adelson, Edward H. Crisp boundary detection using pointwise mutual information. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014.

Kolmogorov, Vladimir and Rother, Carsten. Minimizing nonsubmodular functions with graph cuts - a review. *IEEE transactions on pattern analysis and machine intelligence*, 29(7):1274–9, 2007.

Poon, Hoifung and Domingos, Pedro. Sum-product networks: A new deep architecture. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, pp. 337–346. AUAI Press, 2011.

Shotton, Jamie, Winn, John, Rother, Carsten, and Criminisi, Antonio. TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. *Proceedings European Conference on Computer Vision (ECCV)*, 3951(Chapter 1):1–15, 2006.

Zhao, Yibiao and Zhu, Song-Chun. Image parsing via stochastic scene grammar. In *Advances in Neural Information Processing Systems*, pp. 1–9, 2011.