MODELING AND ESTIMATION OF SPATIAL RANDOM TREES WITH APPLICATION TO IMAGE CLASSIFICATION

I. Pollak, J.M. Siskind, M.P. Harper, and C.A. Bouman

Purdue University
School of Electrical and Computer Engineering
West Lafayette, IN 47907

ABSTRACT

A new class of multiscale multidimensional stochastic processes called spatial random trees is introduced. The model is based on multiscale stochastic trees with stochastic structure as well as stochastic states. Procedures are developed for exact likelihood calculation, MAP estimation of the process, and estimation of the parameters of the process. The new framework is illustrated through a simple binary image classification problem.

1. INTRODUCTION

In this work, we develop a new class of multiscale stochastic models for multidimensional signals that we call *spatial random trees* (SRTs). Similarly to [1, 2], our models are stochastic processes on trees with each leaf corresponding to a single sample. Our key innovation, however, is that the tree structure itself is random and is generated by a *probabilistic grammar* [3].

Probabilistic grammars have been widely used in natural-language processing, for example, to model the structure of sentences [4]. The concept of probabilistic grammar is based on the notion of branching stochastic processes which have been used in studying population dynamics since 1845 [5–7]. These problems have been posed either in 1-D where the objects under consideration, for example, words in sentences, are arranged linearly; or even in "0-D" where the arrangement of objects, such as molecules of different types in a population of particles, does not matter. Recently, there have been efforts to apply probabilistic grammars to 2-D problems such as optical character recognition [8].

These developments have motivated SRTs–our new general framework for modeling multidimensional signals with probabilistic grammars. This framework is described in Section 2 and is the central contribution of this paper. For simplicity, we restrict our exposition of SRTs to 2-D, but the generalization to an arbitrary number of dimensions is straightforward.

With our framework, we obtain exact algorithms for performing the three fundamental tasks required of such models: computing data likelihoods; finding the MAP estimate of both the tree structure and the tree states; and computing the parameter updates required for each iteration of the EM algorithm [9] used to train the model. These resulting algorithms–collectively termed the Center-Surround algorithm–are described in Section 3. They are an extension of–and were inspired by–the Forward-Backward algorithm [4] for hidden Markov models and the Inside-Outside algorithm [4, 10, 11] for 1-D probabilistic grammars.

While extensive experiments with real data are beyond the scope of this paper, we include a simple synthetic example in Section 4 which illustrates our framework.

This work was supported in part by a National Science Foundation (NSF) CAREER award CCR-0093105, a Purdue Research Foundation grant, a Xerox Foundation grant, NSF grants 9987576 and 9980054. Part of this work was carried out while the third author was on leave at NSF. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the view of NSF.

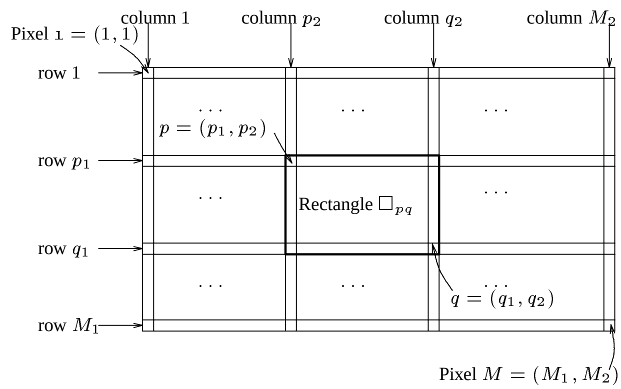

Fig. 1. An illustration of our notation for images.

2. SPATIAL RANDOM TREES

We consider images defined on an $M_1 \times M_2$ rectangular domain illustrated in Fig. 1. In other words, an image $\mathbf{u}$ is an $M_1 \times M_2$ matrix of numbers. The rectangular domain whose upper left corner is $p = (p_1, p_2)$ and whose lower right corner is $q = (q_1, q_2)$ is denoted $\Box_{pq}$. For $p = (p_1, p_2)$, we write u_p and $\Box_{pp} = \Box_p = p$ to denote the value and location, respectively, of the pixel at the intersection of row p_1 and column p_2. We abbreviate $\imath = (1, 1)$ and $M = (M_1, M_2)$, so that the whole domain of definition of image $\mathbf{u}$ is $\Box_{\imath,M}$.

2.1. Probabilistic Grammars and Spatial Random Trees

SRTs model images with binary (dyadic) trees whose leaves are image pixel locations, as illustrated in Fig. 2(a,b). A sample path of an SRT is a (deterministic) tree, i.e. a triple $(\mathcal{V}, \mathcal{E}, x)$ consisting of a set $\mathcal{V}$ of all vertices, a set $\mathcal{E}$ of all edges, and a mapping x which associates a *state* x_α to every vertex α. We distinguish between two types of states: the states corresponding to the image pixel values which can only appear at the leaf vertices of the

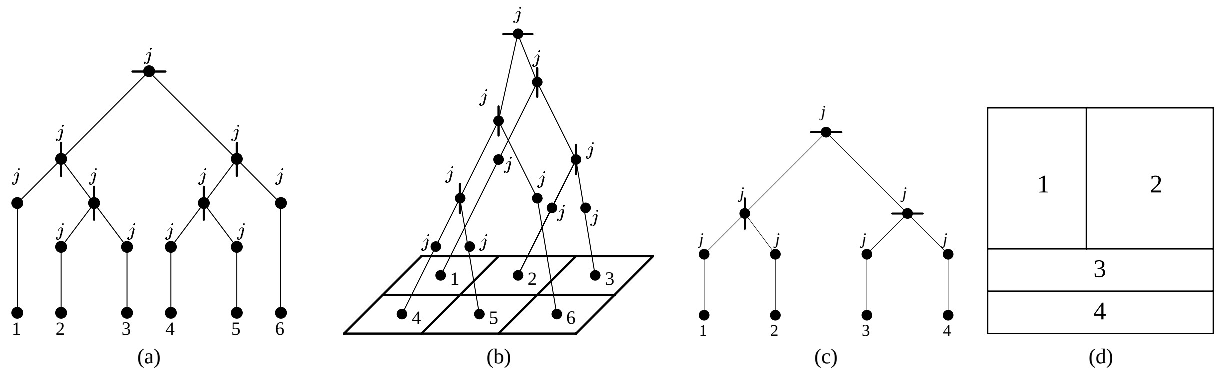

Fig. 2. (a) A tree generated by our image grammar, by applying productions $j \xrightarrow{o} j, j$ and $j \to u$ for $o \in \{h, v\}$ and $u \in \{1, 2, 3, 4, 5, 6\}$. (b) The same tree superimposed onto the corresponding image. A short horizontal (vertical) line through a vertex signifies a horizontal (vertical) split at that vertex. (c) A tree that does not correspond to an $M_1 \times M_2$ rectangular grid. (d) Regardless of the split locations, the first row will have two "pixels" whereas the second and third rows will only have one "pixel" each.

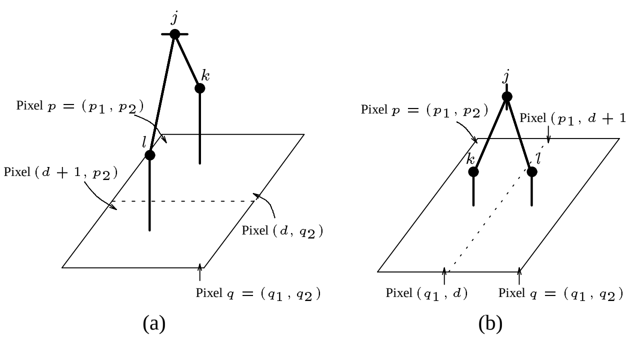

Fig. 3. Possible relationships between the yield of a vertex and the yields of its children: (a) horizontal split; (b) vertical split.

tree, and the "hidden" states corresponding to the remaining vertices of the tree. Any state which can occur at a leaf vertex (i.e. any possible pixel value) is called a *terminal state*, and the set of all terminal states is denoted by $\mathcal{T}$. Any possible state for an internal vertex (i.e. a vertex which is not a leaf) is called a *nonterminal state*, and the set of all nonterminal states is denoted by $\mathcal{N}$.

The *yield* of any internal vertex α, denoted $\mathcal{Y}(\alpha)$, is the set of all leaf descendants of α. In our model, the yield of every internal vertex of a tree is a rectangular region of the image. Every internal vertex whose yield is a single pixel $\Box_p$ is required to have a single child–pixel location $\Box_p$–with a terminal state which is the image value at that pixel, u_p. If the parent of $\Box_p$ has state j, we describe this transition as $j \to u_p$. Following the terminology of natural-language processing, we call any transition of the form $j \to u$ with $j \in \mathcal{N}$ and $u \in \mathcal{T}$, a *terminal production*.

We moreover impose that unless the yield $\Box_{pq}$ of an internal vertex α is a single pixel, α must have two children which are internal vertices with disjoint yields such that the union of the yields is equal to the yield of α. In this case, one further restriction is that the two children be an ordered pair, with the upper left corner $\Box_p$ falling into the yield of the first child and the lower right corner $\Box_q$ falling into the yield of the second child. An equivalent explanation of these requirements is that there are the following possibilities for the yields of the children β and γ of α:

(i) $\mathcal{Y}(\beta) = \Box_{p,(d,q_2)}$ and $\mathcal{Y}(\gamma) = \Box_{(d+1,p_2),q}$ for some $d \in \{p_1, \ldots, q_1 - 1\}$, as illustrated in Fig. 3(a).

(ii) $\mathcal{Y}(\beta) = \Box_{p,(q_1,d)}$ and $\mathcal{Y}(\gamma) = \Box_{(p_1,d+1),q}$ for some $d \in \{p_2, \ldots, q_2 - 1\}$, as illustrated in Fig. 3(b).

If $x_\alpha = j$, $x_\beta = k$, and $x_\gamma = \ell$, we denote a transition of the first type (splitting of $\mathcal{Y}(\alpha)$ along a horizontal line) by $j \xrightarrow{h} k, \ell$ and call it a *horizontal nonterminal production*. We denote a transition of the second type (splitting of $\mathcal{Y}(\alpha)$ along a vertical line) by $j \xrightarrow{v} k, \ell$ and call it a *vertical nonterminal production*. We use $\mathcal{O}$ to denote the set of possible orientations of a nonterminal production: $\mathcal{O} = \{h, v\}$, and we use $\mathcal{P}$ to denote the set of all possible productions (both terminal and nonterminal).

The triple $(\mathcal{N}, \mathcal{T}, \mathcal{P})$ is called a *grammar*. The discussion above means that, in our model, $\mathcal{P}$ consists of the following productions:

$$j \xrightarrow{o} k, \ell \quad \forall j, k, \ell \in \mathcal{N},\ \forall o \in \mathcal{O} \qquad (1)$$

$$j \to u \quad \forall j \in \mathcal{N},\ \forall u \in \mathcal{T}. \qquad (2)$$

Each nonterminal production $j \xrightarrow{o} k, \ell$ is assigned probability $\mathbf{P}_{prod}(j \xrightarrow{o} k, \ell)$, and each terminal production $j \to u$ is assigned probability $\mathbf{P}_{prod}(j \to u)$, in such a way that the following normalization equations are satisfied:

$$\sum_{o,k,l} \mathbf{P}_{prod}(j \xrightarrow{o} k, \ell) + \sum_{u} \mathbf{P}_{prod}(j \to u) = 1, \quad \forall j \in \mathcal{N}.$$

In our model, the state of the root vertex can be any nonterminal state $j \in \mathcal{N}$ with probability $\mathbf{P}_{root}(j)$ where

$$\sum_{j \in \mathcal{N}} \mathbf{P}_{root}(j) = 1.$$

The probability of any tree T is then defined to be the product of the root state probability and the probabilities of all the productions that are involved in generating T. Denoting the set of all internal vertices of T by $\mathcal{V}_{int}$, the root of T by ρ, and the production applied at α by Λ_α, we have:

$$\mathbf{P}(T) \triangleq \mathbf{P}_{root}(x_\rho) \prod_{\alpha \in \mathcal{V}_{int}} \mathbf{P}_{prod}(\Lambda_\alpha).$$

Definition 1. *The stochastic process defined by the probabilistic grammar with productions (1,2), is called a* ***spatial random tree*** *(SRT).*

2.2. Generating Images from the Grammar of Eqs. (1,2)

Note that a sequence of productions from Eqs. (1,2) may generate a tree whose leaves are not arranged in an $M_1 \times M_2$ rectangle. For example, while the tree of Fig. 2(c) is consistent with Eqs. (1,2), the "image" it produces is not defined on an $M_1 \times M_2$ rectangular grid, for any M_1 and M_2. In addition to some desired trees such as the tree of Fig. 2(a), our grammar generates undesired trees. It is moreover unclear whether there may be several images corresponding to the same desired tree. In the previous section, we defined a probability for each tree; what we would like, however, is a probabilistic model for images. We therefore need to resolve the issue of unambiguously associating an image with every tree.

Fortunately it turns out that if a tree does produce an image, that image is unique.

Definition 2 (Admissible trees). *Let T be a tree generated by the grammar of Eqs. (1,2). Let $\mathcal{V}_{int}$ be the set of its internal vertices, and let ρ be its root vertex. Suppose that there exist a pair of positive integers $M = (M_1, M_2)$, and a bijective function*

$$\mathfrak{F} : \mathcal{Y}(\rho) \to \square_{1,M}$$

which uniquely maps each leaf of the tree to a location in an $M_1 \times M_2$ grid, and which has the following property.

> *The yield of each internal vertex of the tree is mapped to a rectangular region. More formally,*
>
> $$\forall \alpha \in \mathcal{V}_{int} \quad \exists p, q \text{ such that } \{\mathfrak{F}(\beta) | \beta \in \mathcal{Y}(\alpha)\} = \square_{pq}.$$

*We then say that T is an **admissible tree**, and $\mathfrak{F}$ is an associated **admissibility function**.*

The following theorem, which we state without proof, shows that if the yield of a tree can be mapped to an image grid in a manner described above and illustrated in Fig. 2(a,b), such a mapping is unique.

Theorem 1 (Admissibility Theorem). *If a tree T is admissible, there is a unique admissibility function for T.*

2.3. Probability Model for Images

Suppose now that we have an $M_1 \times M_2$ image $\mathbf{u} = \mathbf{u}_{1,M}$, and an admissible tree T. If the yield of T is $\square_{1,M}$ and the states of the leaves are $\mathbf{u}_{1,M}$, we say that the tree T *generates* the image $\mathbf{u}$. We define the event $\Omega_{\mathbf{u}}$ to be the set of all admissible trees that generate the image $\mathbf{u}$. The term *probability of image* $\mathbf{u}$ (denoted $\mathbf{P}(\mathbf{u})$) is shorthand for the probability of the set $\Omega_{\mathbf{u}}$. Note that $\mathbf{P}(\mathbf{u})$ does not, in general, define a probability distribution on the set of all images since the set of all admissible trees is not required to have unit probability.

3. SPATIAL RANDOM TREES AND INFERENCE

Our framework of SRTs admits recursive algorithms for likelihood calculation and for the estimation of the MAP (maximum a posteriori probability) tree. The EM algorithm [9] can moreover be adapted to search for the parameter values which maximize the likelihood of an image or a set of images. These algorithms are collectively termed the ***Center-Surround algorithm***. The Center-Surround algorithm is based on recursive calculations involving *center* and *surround* probabilities which we presently describe.

For every rectangular region $\square_{pq}$ of an image $\mathbf{u}$, we define the center probability c^j_{pq} to be the probability of all admissible trees that generate the subimage $\mathbf{u}_{pq}$ and whose root state is j. In other words, the center probability is the conditional probability of subimage $\mathbf{u}_{pq}$ given the event Ω^j where Ω^j is the set of all trees with root state j: $c^j_{pq} = \mathbf{P}(\mathbf{u}_{pq}|\Omega^j)$. In particular, the conditional probability of the whole image given Ω^j is $c^j_{1,M}$. Therefore, the probability of image $\mathbf{u}$ can be easily computed if the center probabilities $c^j_{1,M}$ are known for all possible root states $j \in \mathcal{N}$:

$$\mathbf{P}(\mathbf{u}) = \sum_{j \in \mathcal{N}} c^j_{1,M} \mathbf{P}_{root}(j). \tag{3}$$

The following proposition, illustrated in Fig. 3, gives a recursive algorithm for the computation of $c^j_{1,M}$. It takes advantage of the fact that any center probability for a rectangle containing multiple pixels can be expressed in terms of the center probabilities for smaller rectangles. Note that the first term of the recursion formula below corresponds to summing over all possible horizontal splittings (Fig. 3(a)), and the second term corresponds to the vertical splittings (Fig. 3(b)).

Proposition 1. *For any nonempty rectangular domain $\square_{pq} \subset \square_{1,M}$ with $p \neq q$, and any $j \in \mathcal{N}$,*

$$\begin{aligned} c^j_{pq} \;=\; & \sum_{d=p_1}^{q_1-1} \sum_{k \in \mathcal{N}} \sum_{\ell \in \mathcal{N}} \mathbf{P}_{prod}(j \xrightarrow{h} k, \ell) c^k_{p,(d,q_2)} c^\ell_{(d+1,p_2),q} \\ + & \sum_{d=p_2}^{q_2-1} \sum_{k \in \mathcal{N}} \sum_{\ell \in \mathcal{N}} \mathbf{P}_{prod}(j \xrightarrow{v} k, \ell) c^k_{p,(q_1,d)} c^\ell_{(p_1,d+1),q}. \end{aligned}$$

For any $p \in \square_{1,M}$ and any $j \in \mathcal{N}$,

$$c^j_{pp} = \mathbf{P}_{prod}(j \to u_p).$$

Combining Proposition 1 with Eq. (3) gives a recursive algorithm for calculating the probability $\mathbf{P}(\mathbf{u})$ of image $\mathbf{u}$.

The probability $\mathbf{P}(\mathbf{u})$ can also be recursively calculated using the surround probabilities s^j_{pq}. Each surround probability gives the probability of the image region surrounding $\square_{pq}$. The combination of these two recursions makes it possible to perform one iteration of the EM procedure for estimating the parameters of the SRT from data. Due to space constraints, we are unable to describe the details of the training algorithm in this paper. It will be published elsewhere.

There also exists a dynamic programming algorithm for MAP tree estimation–i.e. for extracting the most probable tree $\Omega_{\mathbf{u}}$ for a given image $\mathbf{u}$. The recursive formulas are a simple variant of the center recursion of Proposition 1, with "$\sum$" replaced by "$\max$". The probability of the most probable tree in $\Omega_{\mathbf{u}}$ with root state j is denoted g^j_{pq}. The base case is:

$$g^j_{pp} = \mathbf{P}_{prod}(j \to u_p).$$

We recursively calculate g^j_{pq} for any rectangle in terms of probabilities associated with smaller rectangles:

$$\begin{aligned} g^{j;h}_{pq} &= \max_{k,\ell,d} \mathbf{P}_{prod}(j \xrightarrow{h} k, \ell)\, g^k_{p,(d,q_2)}\, g^\ell_{(d+1,p_2),q}, \\ g^{j;v}_{pq} &= \max_{k,\ell,d} \mathbf{P}_{prod}(j \xrightarrow{v} k, \ell)\, g^k_{p,(q_1,d)}\, g^\ell_{(p_1,d+1),q}, \\ g^j_{pq} &= \max(g^{j;v}_{pq}, g^{j;h}_{pq}). \end{aligned}$$

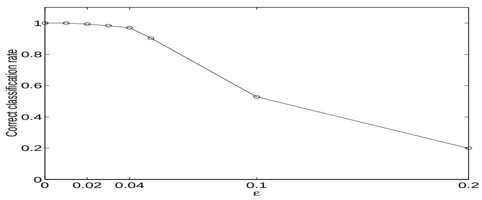

Fig. 4. The rate of correct classification of noisy digit images from the X WINDOWS `9x15` font, as a function of the noise level ε.

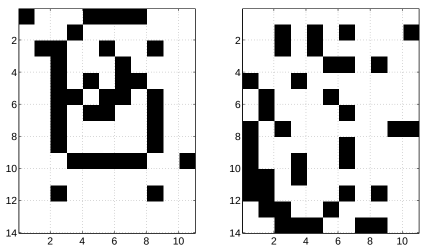

Fig. 5. These two noisy digits (6 and 0) are classified correctly with our algorithm. The noise levels are $\varepsilon = 0.1$ and $\varepsilon = 0.2$, respectively.

We in addition store the four-tuple of parameters (o, k, l, d) which have led to the maximal g^j_{pq} where $o \in \{h, v\}$ stands for the split orientation. We call this four-tuple f^j_{pq}. If $g^{j;h}_{pq} > g^{j;v}_{pq}$, then

$$f^j_{pq} = (h,\ \arg\max_{k,\ell,d} \mathbf{P}_{prod}(j \xrightarrow{h} k, \ell)\, g^k_{p,(d,q_2)}\, g^\ell_{(d+1,p_2),q}).$$

Otherwise,

$$f^j_{pq} = (v,\ \arg\max_{k,\ell,d} \mathbf{P}_{prod}(j \xrightarrow{v} k, \ell)\, g^k_{p,(q_1,d)}\, g^\ell_{(p_1,d+1),q}).$$

If the maximum is not unique, we can choose an arbitrary maximizing four-tuple. The probability of the MAP tree $\widehat{T}$ is calculated from the g variables,

$$\mathbf{P}(\widehat{T}) = \max_{j \in \mathcal{N}} g^j_{1,M} \mathbf{P}_{root}(j),$$

and $\widehat{T}$ itself is constructed from the list of the f variables.

4. EXPERIMENTAL EXAMPLE

We now apply our likelihood computation algorithm of Section 3 to classifying binary images of noisy digits. Our data set consists of the ten digits from the X WINDOWS `9x15` font whose characters are 10×7 pixel images, placed at various locations on a white 14×11 background. These images are corrupted by synthetic noise which independently flips every pixel with probability ε.

For each level of noise ε and each digit $k = 0, 1, \ldots, 9$, a probabilistic grammar $\mathcal{G}_{k,\varepsilon}$ was obtained through a combination of automatic training via the EM algorithm and manually writing down certain productions and their probabilities. For several noise levels in the range $0 \le \varepsilon \le 0.2$, we conducted 900 classification experiments with noisy digit images. Each of the 900 images was classified by calculating its likelihoods with respect to the ten grammars $\mathcal{G}_{0,\varepsilon}, \ldots, \mathcal{G}_{9,\varepsilon}$. Classifying each image took about 3 seconds on an 800 MHz Pentium III processor.

Our experiments are summarized in Fig. 4 which shows a plot of our estimates of the correct classification probability as a function of the noise level ε, from the noise-free case $\varepsilon = 0$ to the extremely noisy case of $\varepsilon = 0.2$. This latter case corresponds to an average of about 31 incorrect pixels per 14×11 image, which, as shown in Fig. 5, makes some images difficult to recognize for a human. The plot in Fig. 4 demonstrates excellent performance of our algorithm and graceful degradation for very noisy images.

5. CONCLUSIONS

We have presented general methods for computing the likelihood of an observation of a multidimensional random field, and for estimating both the structure and the states of a stochastic tree from such an observation. We refer to the associated new class of multiscale processes as spatial random trees. These models can be used to classify and interpret images, and they can be trained using the EM algorithm. A simple experiment illustrates their potential value in signal-processing applications.

6. ACKNOWLEDGMENTS

We would like to thank Yan Huang, Bill Nagel, James Sherman, and Eric Théa for developing the code for the Center-Surround algorithm and running the experiment of Section 4.

7. REFERENCES

[1] M. Basseville, A. Benveniste, K. C. Chou, S. A. Golden, R. Nikoukhah, and A. S. Willsky, "Modeling and estimation of multiresolution stochastic processes," *IEEE Trans. on Information Theory*, vol. 38, no. 2, pp. 766–784, March 1992.

[2] C. A. Bouman and M. Shapiro, "A multiscale random field model for Bayesian image segmentation," *IEEE Trans. on Image Processing*, vol. 3, no. 2, pp. 162–177, March 1994.

[3] D. Sankoff, "Branching processes with terminal types: application to context-free grammars," *Journal of Applied Probability*, vol. 8, pp. 233–240, 1971.

[4] C. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[5] I. J. Bienaymé, "De la loi de multiplication et de la durée des familles," *Soc. Philomat. Paris Extraits, Sér.*, vol. 5, pp. 37–39, 1845. Reprinted as an Appendix to [7].

[6] H. W. Watson and F. Galton, "On the probability of extinction of families," *Journal of the Anthropological Institute of Great Britain and Ireland*, vol. 4, pp. 138–144, 1875.

[7] D. G. Kendall, "The genealogy of genealogy: Branching processes before (and after) 1873," *Bull. London Math. Soc.*, vol. 7, pp. 225–253, 1975.

[8] D. Potter, *Compositional Pattern Recognition*. PhD thesis, Brown University, 1999. `http://www.dam.brown.edu/people/dfp.`

[9] L. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Ann. Math. Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[10] J. Baker, "Trainable grammars for speech recognition," *Speech Communications Papers for the 97th Meeting of the Acoust. Soc. of America* (D. Klatt and J. Wolf, eds.), 1979.

[11] K. Lari and S. Young, "The estimation of stochastic context-free grammars using the inside-outside algorithm," *Computer Speech and Language*, vol. 4, pp. 35–56, 1990.