PARAMETER ESTIMATION FOR SPATIAL RANDOM TREES USING THE EM ALGORITHM

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 was recently introduced in [9]. The model is based on multiscale stochastic trees with stochastic structure as well as stochastic states. In this work, we describe a method for estimating the parameters of the process.

1. INTRODUCTION

In [9], we developed a new class of multiscale stochastic models for multidimensional signals called *spatial random trees* (SRTs). Similarly to [2, 5], 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 context-free grammar* [11].

Probabilistic grammars have been widely used in natural-language processing, for example, to model the structure of sentences [8]. The concept of probabilistic grammar is based on the notion of branching stochastic processes which have been used in studying population dynamics since 1845 [4, 6, 12]. 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 [10].

These developments have motivated SRTs–our framework for modeling multidimensional signals with probabilistic grammars. For simplicity, we restrict our exposition of SRTs to 2-D, but the generalization to an arbitrary number of dimensions is straightforward. Our framework is reviewed in Section 2. The inference algorithms for our framework are collectively termed the Center-Surround algorithm and were inspired by the Forward-Backward algorithm [8] for hidden Markov models and the Inside-Outside algorithm [1, 7, 8] for 1-D probabilistic context-free grammars. We described two components of the Center-Surround algorithm in [9], specifically, the exact algorithms for computing data likelihoods and finding the MAP estimate of both the tree structure and the tree states. In Section 3 of the present paper, we describe an exact algorithm for computing the parameter updates required for each iteration of the EM algorithm [3] used to train the model. The proofs of our results cannot be included in this paper due to space constraints and will be published elsewhere. Experimental examples in Section 4 illustrate our framework.

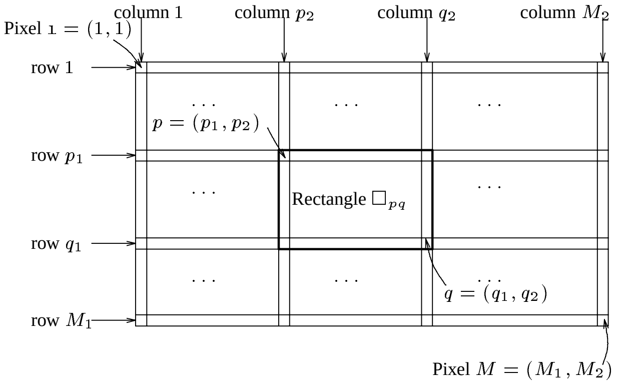


Fig. 1. An illustration of our notation for images.

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 $\square_{pq}$. For $p = (p_1, p_2)$, we write u_p and $\square_{pp} = \square_p = p$ 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 $\square_{\imath,M}$.

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 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 $\mathcal{N}$.

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.

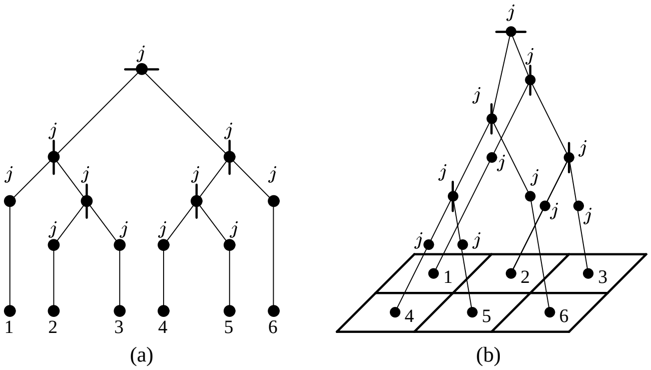


Fig. 2. (a) A tree generated by our image grammar, by applying productions $j \overset{o}{\to} j, j$ and $j \to u$ for $o \in \{h, v\}$ and $u \in \{1, 2, 3, 4, 5, 6\}$. (b) The same tree superimposed onto the corresponding image. A short horizontal (vertical) line through a vertex signifies a horizontal (vertical) split at that vertex.

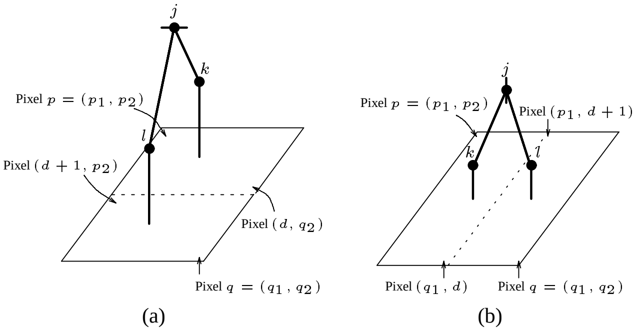


Fig. 3. Possible relationships between the yield of a vertex and the yields of its children: (a) horizontal split; (b) vertical split.

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 $\square_p$ is required to have a single child–pixel location $\square_p$–with a terminal state which is the image value at that pixel, u_p. If the parent of $\square_p$ 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 $\square_{pq}$ 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 $\square_p$ falling into the yield of the first child and the lower right corner $\square_q$ 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) $\mathcal{Y}(\beta) = \square_{p,(d,q_2)}$ and $\mathcal{Y}(\gamma) = \square_{(d+1,p_2),q}$ for some $d \in \{p_1, \ldots, q_1 - 1\}$, as illustrated in Fig. 3(a).

(ii) $\mathcal{Y}(\beta) = \square_{p,(q_1,d)}$ and $\mathcal{Y}(\gamma) = \square_{(p_1,d+1),q}$ 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 $j \overset{h}{\to} k, \ell$ 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 $j \overset{v}{\to} k, \ell$ 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:

$$j \overset{o}{\to} k, \ell \quad \forall j, k, \ell \in \mathcal{N},\ \forall o \in \mathcal{O} \tag{1}$$

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

Each nonterminal production $j \overset{o}{\to} k, \ell$ is assigned probability $\mathbf{P}_{prod}(j \overset{o}{\to} k, \ell)$, 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:

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

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****.*

As discussed in [9], a sequence of productions from Eqs. (1,2) may generate a tree whose leaves cannot be arranged in an $M_1 \times M_2$ rectangle. It turns out, however, that if a tree's leaves can be associated with an $M_1 \times M_2$ rectangle, such association is unique. If the yield of a tree T forms a rectangle $\square_{1,M}$, and the states of the leaves are $\mathbf{u}$, then we say that the tree T *generates* the image $\mathbf{u}$. We define the event $\Omega_{\mathbf{u}}$ to be the set of all 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 $\Omega_{\mathbf{u}}$.

3. PARAMETER ESTIMATION VIA EM

Our framework of SRTs admits recursive algorithms for likelihood calculation and for the estimation of the MAP (maximum a posteriori probability) tree. The EM (expectation maximization) algorithm [3] 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 conditional probability of the subimage $\mathbf{u}_{pq}$ given that it is generated by a subtree whose root state is j: $c^j_{pq} = \mathbf{P}(\mathbf{u}_{pq}|\Omega^j)$, where Ω^j is the set of all trees with root state j.

The following proposition, taken from [9] and illustrated in Fig. 3, gives a recursive algorithm for the computation of c^j_{pq}. 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.

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

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

For any $p \in \Box_{1,M}$ and any $j \in \mathcal{N}$, $c^j_{pp} = \mathbf{P}_{prod}(j \to u_p)$.

Each surround probability s^j_{pq} is the joint probability of the image region surrounding $\Box_{pq}$ and the event that the subimage $\mathbf{u}_{pq}$ is generated by a subtree whose root state is j. The surround probabilities can also be calculated recursively; however, these recursions use the center probabilities which must therefore be precomputed.

Proposition 2. *For any nonempty rectangular domain $\Box_{pq} \subset \Box_{1,M}$ with $p \neq q$, and any $j \in \mathcal{N}$,*

$$\begin{aligned} s^j_{pq} &= \sum_{e=1}^{p_1-1} \sum_{k,\ell} s^k_{(e,p_2),q} \mathbf{P}_{prod}(k \xrightarrow{h} \ell, j) c^\ell_{(e,p_2),(p_1-1,q_2)} \\ &+ \sum_{e=q_1+1}^{M_1} \sum_{k,\ell} s^k_{p,(e,q_2)} \mathbf{P}_{prod}(k \xrightarrow{h} j, \ell) c^\ell_{(q_1+1,p_2),(e,q_2)} \\ &+ \sum_{e=1}^{p_2-1} \sum_{k,\ell} s^k_{(p_1,e),q} \mathbf{P}_{prod}(k \xrightarrow{v} \ell, j) c^\ell_{(p_1,e),(q_1,p_2-1)} \\ &+ \sum_{e=q_2+1}^{M_2} \sum_{k,\ell} s^k_{p,(q_1,e)} \mathbf{P}_{prod}(k \xrightarrow{v} j, \ell) c^\ell_{(p_1,q_2+1),(q_1,e)}, \end{aligned}$$

where our convention is that any sum over an empty set is zero. The base case for this recursion is: $s^j_{1,M}(\mathbf{u}) = \mathbf{P}_{root}(j)$.

As the next proposition shows, the combination of the center and surround recursions makes it possible to perform one iteration of the EM procedure for estimating the parameters of the SRT from data. Starting with any initial parameter values $(\mathbf{P}^0_{root}, \mathbf{P}^0_{prod})$, the EM algorithm [3] generates a sequence of parameter estimates $(\mathbf{P}^0_{root}, \mathbf{P}^0_{prod}), (\mathbf{P}^1_{root}, \mathbf{P}^1_{prod}), (\mathbf{P}^2_{root}, \mathbf{P}^2_{prod}), \ldots$ which is guaranteed to climb the likelihood surface. The next proposition gives the EM update equations for our problem.

Proposition 3. *Suppose we have an observation $\mathbf{u}$. Denoting $\mathbf{P}^n_{jk\ell,o} = \mathbf{P}^n_{prod}(j \xrightarrow{o} k, \ell)$, the EM update equations for the parameters of our probabilistic image grammar are:*

$$\begin{aligned} \mathbf{P}^{n+1}_{root}(j) &= \mathbf{P}^n_{root}(j), \\ \mathbf{P}^{n+1}_{jk\ell,h} &= \frac{\sum_{p,q} s^j_{pq} \mathbf{P}^n_{jk\ell,h} \sum_{d=p_1}^{q_1-1} c^k_{p,(d,q_2)} c^\ell_{(d+1,p_2),q}}{\sum_{p,q} s^j_{pq} c^j_{pq}}, \\ \mathbf{P}^{n+1}_{jk\ell,v} &= \frac{\sum_{p,q} s^j_{pq} \mathbf{P}^n_{jk\ell,v} \sum_{d=p_2}^{q_2-1} c^k_{p,(q_1,d)} c^\ell_{(p_1,d+1),q}}{\sum_{p,q} s^j_{pq} c^j_{pq}}, \\ \mathbf{P}^{n+1}_{prod}(j \to u) &= \frac{\mathbf{P}^n_{prod}(j \to u) \sum_{p:u_{pp}=u} s^j_{pp}}{\sum_{p,q} s^j_{pq} c^j_{pq}}, \end{aligned}$$

where all center and surround variables in the righthand sides are calculated using the parameters $\mathbf{P}^n_{prod}$, and where each double summation over p and q is done over all such pairs (p,q) that $\Box_{p,q}$ is a nonempty rectangular subdomain of $\Box_{1,M^i}$. As before, our convention is that any sum over an empty set is zero.

We note that this proposition can be easily modified to account for training on multiple images.

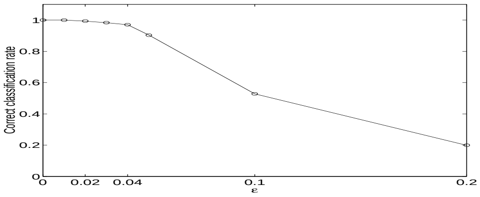


Fig. 4. The rate of correct classification of noisy digit images from the X WINDOWS `9x15` font, as a function of the noise level ε.

4. EXPERIMENTAL EXAMPLES

We now apply our parameter estimation algorithm of Section 3 to two problems involving classification and segmentation of noisy images. For the first experiment, 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 noise-free digit $k = 0, 1, \ldots, 9$, a probabilistic grammar $\mathcal{G}_k$ was obtained from the EM algorithm of Section 3, by training on a single 10×7 image of the digit. Each grammar $\mathcal{G}_k$ was then manually expanded (i.e., several new nonterminal states and nonterminal production rules were introduced) to obtain a new probabilistic grammar capable of placing the 10×7 image of the digit k at any location on a white 14×11 background. For each level of noise ε, the resulting grammar was further manually modified, to account for the noise. Thus, 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.

After the training stage was complete, we conducted 900 classification experiments with noisy digit images for several noise levels in the range $0 \leq \varepsilon \leq 0.2$. Each of the 900 images was classified by calculating its likelihoods with respect to the ten grammars $\mathcal{G}_{0,\varepsilon}, \ldots, \mathcal{G}_{9,\varepsilon}$ and choosing the hypothesis corresponding to the largest likelihood. The likelihood calculation was performed using our algorithm described in [9]. Classifying each image took about 3 seconds on an 800 MHz Pentium III processor.

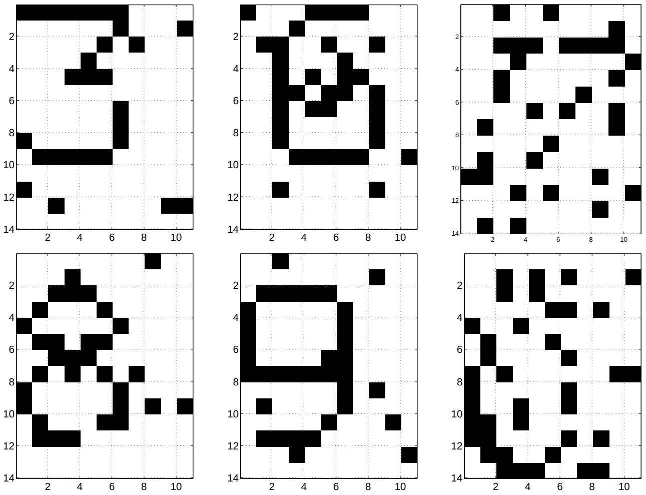
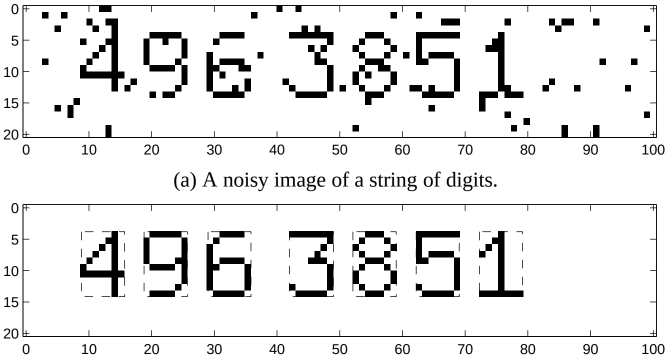


Fig. 5. Six images that were classified correctly. Left: digits 3 and 8, noise level $\varepsilon = 0.05$; center: digits 6 and 9, noise level $\varepsilon = 0.1$; right: digits 7 and 0, noise level $\varepsilon = 0.2$.



(a) A noisy image of a string of digits.

(b) The image and segmentation (dashed lines) produced by our algorithm.

Fig. 6. Given the noisy image shown in (a), our algorithm identifies each of the seven digits correctly and produces the segmentation shown in dashed lines in (b).

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.

In our second example, we use the MAP estimation algorithm of [9] to extract a string of digits from a noisy image and classify these digits. The EM stage of the training process was identical to the one used in the first experiment. The resulting probabilistic grammars $\mathcal{G}_0, \mathcal{G}_1, \ldots, \mathcal{G}_9$ for the ten digits were manually embedded in a larger grammar which describes strings of seven digits on a white background. Similarly to the first experiment, this latter grammar was further manually modified to account for noise. The grammar obtained through this procedure is capable of generating images such as Fig. 6(a). In this grammar, there are ten special nonterminal states $digit$-0, $digit$-1, …, $digit$-9 which are used to label the ten digits. For example, $x_\alpha = digit$-0 is interpreted to mean that digit zero is present in the image and is situated in $\mathcal{Y}(\alpha)$.

Given an image such as that of Fig. 6(a), we use our algorithm of [9] to estimate the MAP tree. For each internal vertex α of this tree such that $x_\alpha = digit\text{-}k$, we extract the rectangle $\mathcal{Y}(\alpha)$ and label it as digit k. Our algorithm therefore produces the segmentation of our image into digits and background, and recognizes each digit. For the input image of Fig. 6(a), this results in Fig. 6(b).

5. CONCLUSIONS

In this paper, we have further analyzed a new class of multiscale stochastic processes called spatial random trees which was introduced in [9]. We have presented general methods for training our models with the help of the EM algorithm, and illustrated their effectiveness through applications to image classification and segmentation.

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] 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.

[2] 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.

[3] 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.

[4] 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 [6].

[5] 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.

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

[7] 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.

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

[9] I. Pollak, J. M. Siskind, M. Harper, and C. A. Bouman, “Modeling and estimation of spatial random trees with application to image classification,” submitted to *ICASSP*, Hong Kong, 2003.

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

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

[12] 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.