Serdica J. Computing **4** (2010), 463–474

**Serdica**
Journal of Computing

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

# USING INSIDE-OUTSIDE ALGORITHM FOR ESTIMATION OF THE OFFSPRING DISTRIBUTION IN MULTITYPE BRANCHING PROCESSES

Nina Daskalova*

Abstract. Multitype branching processes (MTBP) model branching structures, where the nodes of the resulting tree are particles of different types. Usually such a process is not observable in the sense of the whole tree, but only as the "generation" at a given moment in time, which consists of the number of particles of every type. This requires an EM-type algorithm to obtain a maximum likelihood (ML) estimate of the parameters of the branching process. Using a version of the inside-outside algorithm for stochastic context-free grammars (SCFG), such an estimate could be obtained for the offspring distribution of the process.

**1. Introduction.** Multitype branching processes (MTBP) are stochastic models in population dynamics, where particles are of different types.

---

*ACM Computing Classification System* (1998): G.3, F.4.2.

*Key words:* multitype branching processes, offspring distribution, maximum likelihood estimation, expectation maximization, stochastic context-free grammars, inside-outside algorithm.

*The research was partly supported by appropriated state funds for research allocated to Sofia University (contract 112/2010), Bulgaria.

The theory and application of such processes can be found in several books [1, 2, 8, 13]. Statistical inference in MTBP depends on the kind of observation available, whether the whole family tree has been observed, or only the particles existing at given moment $t$, or sometimes even the relative frequencies of types at that moment.

We consider a MTBP $\mathbf{Z}(t) = (Z_1(t), Z_2(t), \dots Z_d(t))$, where $Z_k(t)$ denotes the number of particles of type $T_k$ at time $t$, $k = 1, 2, \dots d$. Some estimators as to whether the entire tree has been observed could be found in [7, 17], but usually we don't have such information about the process. Yakovlev and Yanev in [16] develop some statistical methods for obtaining ML estimators for the offspring characteristics, based on observation on the relative frequencies of types at time $t$. Other approaches use simulation and Monte Carlo methods [6, 9, 10].

When the entire tree is not observed, but only the particles existing at given moment, an Expectation Maximization (EM) algorithm could be used, considering the tree as the hidden data. Such algorithms exist for strictures called Stochastic Context-free Grammars (SCFG). A number of sources point out the relation between MTBPs and SCFGs [5, 15].

SCFGs are used in linguistics and, since recently, in bioinformatics to model the hidden structure of sequences of words or symbols [5, 4]. SCFGs are actually a kind of MTBPs and their properties could be obtained through the theory of branching processes [15]. Our purpose is to use the well developed methods for estimating parameters of SCFGs to estimate offspring distribution probabilities in some MTBPs.

The paper is organized as follows. In Section 2 the EM algorithm is briefly explained. Section 3 shows how an EM algorithm could be constructed to estimate the offspring probabilities of a branching process. In Section 4 the well-known inside-outside algorithm for SCFG is explained. The next Section 5 proposes how this algorithm could be used for branching processes and an example is given at the end of the paper.

**2. The EM Algorithm.** The EM algorithm was explained and given its name in a paper by Dempster, Laird, and Rubin [3]. It is a method for finding maximum likelihood estimates of parameters in statistical models, where the model depends on unobserved latent variables. Let a statistical model be determined by parameters $\theta$, $x$ be the observation and $Y$ be some "hidden" data which determines the probability distribution of $x$. Then the joint probability of the "complete" observation is $P(x, Y|\theta)$ and the probability of the "incomplete" observation is the marginal probability $P(x|\theta) = \sum_y P(x, y|\theta)$. The aim is to maximize the log likelihood

$$\log L(\theta|x) = \log P(x|\theta) = \log \sum_y P(x, y|\theta).$$

For given $\theta^{(i)}$, using $P(x, Y|\theta) = P(Y|x, \theta)P(x|\theta)$ it follows that $\log L(\theta|x) = \log P(x, Y|\theta) - \log P(Y|x, \theta)$, so

$$\log L(\theta|x) = \sum_y P(y|x, \theta^{(i)}) \log P(x, y|\theta) - \sum_y P(y|x, \theta^{(i)}) \log P(y|x, \theta)$$

Write

$$Q(\theta|\theta^{(i)}) = \sum_y P(y|x, \theta^{(i)}) \log P(x, y|\theta).$$

We want the model with parameters $\theta$ to be better than the one with parameters $\theta^{(i)}$, so $\log L(\theta|x) > \log L(\theta^{(i)}|x)$. But

$$\log L(\theta|x) - \log L(\theta^{(i)}|x) = Q(\theta|\theta^{(i)}) - Q(\theta^{(i)}|\theta^{(i)}) + \sum_y P(y|x, \theta^{(i)}) \log \frac{P(y|x, \theta^{(i)})}{P(y|x, \theta)}$$

The last term above is the relative entropy of $P(Y|x, \theta^{(i)})$ relative to $P(Y|x, \theta)$, so it is non-negative and

$$\log L(\theta|x) - \log L(\theta^{(i)}|x) \geq Q(\theta|\theta^{(i)}) - Q(\theta^{(i)}|\theta^{(i)})$$

with equality only if $\theta = \theta^{(i)}$, or if $P(Y|x, \theta^{(i)}) = P(Y|x, \theta)$ for some other $\theta \neq \theta^{(i)}$. Choosing $\theta^{(i+1)} = argmax_\theta Q(\theta|\theta^{(i)})$ will make the difference positive and the likelihood will increase untill a maximum is reached. The *Expectation Maximization Algorithm* is usually stated formally like this:

- *E-step:* Calculate function $Q(\theta|\theta^{(i)})$.

- *M-step:* Maximize $Q(\theta|\theta^{(i)})$ with respect to $\theta$.

More about the theory and applications of the EM algorithm could be found in [12].

**3. The EM Algorithm for MTBP.** Let $x$ be the observed set of particles, $\pi$ is the unobserved tree structure and $\theta$ is the set of parameters—the offspring probabilities. Then the joint probability of the "complete" observation is:

$$P(x,\pi|\theta)=\prod_{\omega}\theta(\omega)^{c(\omega;\pi,x)}=\prod_{T_v\to\mathcal{A}}p(T_v\to\mathcal{A})^{c(T_v\to\mathcal{A};\pi,x)},$$

where $T_v \to \mathcal{A}$ is the rule that a particle of type $T_v$ produces the set of particles $\mathcal{A}$ and $c$ is a count function. We have $\sum_{\mathcal{A}} p(T_v \to \mathcal{A}) = 1$. The probability of the "incomplete" observation is the marginal probability $P(x|\theta) = \sum_{\pi} P(x,\pi|\theta)$. For the EM algorithm we need to compute the function

$$\begin{aligned}
Q(\theta|\theta^{(i)}) &= E_{\theta^{(i)}}(\log P(x,\pi|\theta)) = \sum_{\pi} P(\pi|x,\theta^{(i)})\log P(x,\pi|\theta)\\
&= \sum_{\pi} P(\pi|x,\theta^{(i)}) \sum_{T_v\to\mathcal{A}} c(T_v\to\mathcal{A};\pi,x)\log p(T_v\to\mathcal{A})\\
&= \sum_{T_v\to\mathcal{A}}\sum_{\pi} P(\pi|x,\theta^{(i)})c(T_v\to\mathcal{A};\pi,x)\log p(T_v\to\mathcal{A})\\
&= \sum_{T_v\to\mathcal{A}} E_{\theta^{(i)}}c(T_v\to\mathcal{A})\log p(T_v\to\mathcal{A})
\end{aligned}$$

Taking a partial derivative with respect to $p(T_v \to \mathcal{A})$ and using the Lagrangian multiplier $\sum_{\mathcal{A}} E_{\theta^{(i)}}(T_v \to \mathcal{A}) = \lambda$, we get to the result that the re-estimating parameters are the normalized expected counts

$$p^{(i+1)}(T_v\to\mathcal{A}) = \frac{E_{\theta^{(i)}}c(T_v\to\mathcal{A})}{\sum_{\mathcal{A}} E_{\theta^{(i)}}c(T_v\to\mathcal{A})} = \frac{E_{\theta^{(i)}}c(T_v\to\mathcal{A})}{E_{\theta^{(i)}}c(T_v)}$$

where the expected number of times a particle of type $T_v$ appears in the tree $\pi$ is:

$$E_{\theta^{(i)}}c(T_v) = \sum_{\pi} P(\pi|x,\theta^{(i)})c(T_v;\pi,x).$$

The M-step is explicitly solved, so no effort on maximization is needed. The problem is that, in general, enumerating all possible trees $\pi$ is of exponential complexity. We propose using the inside-outside algorithm for stochastic context-free grammars to reduce complexity.

**4. Estimation in SCFG.** Grammars are a well-developed tool for modelling strings of symbols in computational linguistics. Stochastic grammars give a probabilistic approach to the problems in that field. A stochastic context-free grammar (SCFG) consists of a number of symbols and a number of production rules of the form $\alpha \to \beta$, where $\alpha$ and $\beta$ are sequences of these symbols. The symbols can be of two kinds—abstract nonterminal and terminal that actually appear in an observation. There are also probabilities assigned to the rules. For a SCFG to be in Chomsky normal form it is necessary for the rules to be of the form $X \to YZ$ or $X \to a$, where $X, Y, Z$ are nonterminals and $a$ is a terminal symbol. Every CFG can be represented in Chomsky normal form. For such grammars there exists an EM-type algorithm, called the inside-outside algorithm [11], which finds an ML estimator of the parameters $\theta$ of that grammar, namely the probabilities of the rules, called the transition and emission probabilities respectively for the first and the second type of rules above. It is a three-dimensional dynamic programming algorithm. Let $x$ be the observed sequence of terminals of length $L$, and there be $M$ different nonterminals $W_1$, $W_2$, $\ldots$, $W_M$. Production rules are of the form $W_v \to W_x W_y$ and $W_v \to a$ with transition and emission probabilities $t_v(x, y)$ and $e_v(a)$ respectively. The algorithm consists of three parts—inside, outside and EM re-estimation, which are shown below.

The **Inside** part calculates the probability $\alpha(i, j, v)$ of a parse subtree rooted at nonterminal $W_v$ for subsequence $x_i, \ldots, x_j$ for all $i, j, v$. Formally, it could be written in this way:

- *Initialization*: for $i = 1$ to $L$, $v = 1$ to $N$: $\alpha(i, j, v) = e_v(x_i)$.

- *Iteration*: for $i = 1$ to $L - 1$, $j = i + 1$ to $L$, $v = 1$ to $N$: $\alpha(i, j, v) = \sum_{y=1}^{N} \sum_{z=1}^{N} \sum_{k=i}^{j-1} \alpha(i, k, y)\alpha(k + 1, j, z)t_v(y, z)$.

- *Termination*: $P(x|\theta) = \alpha(1, L, 1)$.

The **Outside** part calculates the probability $\beta(i, j, v)$ of a complete parse subtree rooted at the start nonterminal for the complete sequence $x$, excluding subsequence $x_i, \ldots, x_j$ rooted at nonterminal $W_v$ for all $i, j, v$.

- *Initialization*: $\beta(1, L, 1) = 1$; for $v = 2$ to $N$: $\beta(1, L, v) = 0$.

- *Iteration*: for $i = 1$ to $L$, $j = L$ to $i$, $v = 1$ to $N$: $\beta(i, j, v) = \sum_{y=1}^{N} \sum_{z=1}^{N} \sum_{k=1}^{i-1} \alpha(k, i - 1, z)\beta(k, j, y)t_y(z, v)$

$+\sum_{y=1}^{N}\sum_{z=1}^{N}\sum_{k=j+1}^{L}\alpha(j+1,k,z)\beta(i,k,y)t_y(v,z)$.

- *Termination*: $P(x|\theta)=\sum_{v=1}^{N}\beta(i,i,v)e_v(x_i)$ for any $i$.

For every $W_v \in \pi$ the expected number of times $c(W_v)$ that $W_v$ is used in the tree $\pi$ could be presented as follows:

$$E_\theta c(W_v)=\sum_{\pi}P(\pi|x,\theta)c(W_v;\pi,x)=\sum_{\pi}\frac{P(x,\pi|\theta)}{P(x|\theta)}c(W_v;\pi,x)$$

$$=\frac{1}{P(x|\theta)}\sum_{\pi}P(x,\pi|\theta)c(W_v;\pi,x)=\frac{1}{P(x|\theta)}\sum_{\pi:W_v\in\pi}P(x,\pi|\theta)$$

$$=\frac{1}{P(x|\theta)}\sum_{i}\sum_{j}\alpha(i,j,v)\beta(i,j,v),$$

where $\alpha(i,j,v)$ and $\beta(i,j,v)$ are the inside and outside probabilities for observation $x$.

Similarly, the expected number of times a rule $W_v \to W_yW_z$ is used could be calculated:

$$E_\theta c(W_v\to W_yW_z)=$$

$$\frac{1}{P(x|\theta)}\sum_{i=1}^{L-1}\sum_{j=i+1}^{L}\sum_{k=i}^{j-1}\beta(i,j,v)\alpha(i,k,z)\alpha(k+1,j,z)t_v(y,v),$$

and the expectation for the rule $W_v \to a$ is:

$$E_\theta c(W_v\to a)=\frac{1}{P(x|\theta)}\sum_{i|x_i=a}\beta(i,i,v)e_v(a).$$

Dividing the expectations above, we obtain the **EM re-estimation** of the parameters:

$$t_v^{(n+1)}(y,z)=\frac{\sum\limits_{i=1}^{L-1}\sum\limits_{j=i+1}^{L}\sum\limits_{k=i}^{j-1}\beta(i,j,v)\alpha(i,k,z)\alpha(k+1,j,z)t_v^{(n)}(y,v)}{\sum\limits_{i=1}^{L}\sum\limits_{j=i}^{L}\alpha(i,j,v)\beta(i,j,v)}$$

$$e_v^{(n+1)}(a) = \frac{\sum\limits_{i|x_i=a} \beta(i,i,v) e_v^{(n)}(a)}{\sum\limits_{i=1}^{L} \sum\limits_{j=i}^{L} \alpha(i,j,v)\beta(i,j,v)}$$

For several observed sequences the expected numbers in the nominator and denominator are summed up for all sequences.

The time complexity of the algorithm is $O(L^3N^3)$.

**5. MTBP as a SCFG and using the Inside-Outside Algorithm for MTBP.** An MTBP could be represented as a SCFG in the following way. First our process has to be represented only with “rules”” of the form

$$X \xrightarrow{p} \{Y, Z\},$$

which means that a particle of type $X$ could produce two particles of types $Y$ and $Z$ with probability $p$. For every such rule in the process, the corresponding SCFG will include nonterminals $\{X, Y, Z, Y^T, Z^T\}$, terminals $\{y, z\}$ and rules

$$X \xrightarrow{p_1} YZ|ZY, \quad X \xrightarrow{p_2} Y^TZ|ZY^T, \quad X \xrightarrow{p_3} YZ^T|Z^TY,$$

$$X \xrightarrow{p_4} Y^TZ^T|Z^TY^T, \quad Y^T \xrightarrow{1} y, \quad Z^T \xrightarrow{1} z,$$

and $p_1 + p_2 + p_3 + p_4 = p$.

Here $Y^T$ and $Z^T$ are nonterminals of “terminal” type, meaning that they transform into terminals $y$ and $z$ only. We regard these terminals as the observed particles, and the other nonterminals represent the hidden structure of the process. Thus for a single rule in the process there are six rules in the grammar and the number of types doubles.

To use the Inside-Outside Algorithm for MTBP, we take the following steps:

1. Construct the corresponding SCFG.
2. Estimate parameters for SCFG using as observed sequences all possible permutations of the observed set of particles. Thus, if we have observed 2 particles of type $X$ and 1 of type $Y$, we use as “observed sequences” all $xxy$, $xyx$ and $yxx$.

3. If the number of permutations is large, a Monte Carlo sample approach could be used to obtain the estimate.

4. Calculate probabilities in MTBP summing up the ones estimated in SCFG.

**6. Examples.** We consider an MTBP with three types of particles $T_1$, $T_2$ and $T_3$, where the third type is terminal—a particle of this type does not reproduce, and for the other two types all productions are allowed:

$$T_i \to \{T_1, T_2\}, \quad T_i \to \{T_1, T_3\}, \quad T_i \to \{T_2, T_3\},$$

$$T_i \to \{T_1, T_1\}, \quad T_i \to \{T_2, T_2\}, \quad T_i \to \{T_3, T_3\},$$

for $i = 1, 2$.

The corresponding SCFG has nonterminals $T_1$, $T_2$, $T_3$, $T^T_1$, $T^T_2$ and $T^T_3$, terminals $t_1$, $t_2$ and $t_3$, and rules:

$$T_1 \to T_1 T_2 | T_2 T_1, \quad T_1 \to T^T_1 T_2 | T_2 T^T_1,$$

$$T_1 \to T_1 T^T_2 | T^T_2 T_1, \quad T_1 \to T^T_1 T^T_2 | T^T_2 T^T_1,$$

$$T^T_1 \xrightarrow{1} t_1, \quad T^T_2 \xrightarrow{1} t_2,$$

Table 1. Estimation for the parameters of the SCFG based on all permutations (140)

| $T_1$: | $T_1$ | $T_2$ | $T_3$ | $T^T_1$ | $T^T_2$ | $T^T_2$ |
|---|---|---|---|---|---|---|
| $T_1$ | 0.0000 | 0.0600 | 0.0000 | 0.0000 | 0.1900 | 0.0000 |
| $T_2$ | 0.0600 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0650 |
| $T_3$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $T^T_1$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $T^T_2$ | 0.1900 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.1850 |
| $T^T_3$ | 0.0000 | 0.0650 | 0.0000 | 0.0000 | 0.1850 | 0.0000 |

| $T_2$: | $T_1$ | $T_2$ | $T_3$ | $T^T_1$ | $T^T_2$ | $T^T_2$ |
|---|---|---|---|---|---|---|
| $T_1$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.2500 |
| $T_2$ | 0.0000 | 0.0000 | 0.0000 | 0.2500 | 0.0000 | 0.0000 |
| $T_3$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $T^T_1$ | 0.0000 | 0.2500 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $T^T_2$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $T^T_3$ | 0.2500 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

for the first rule above and similarly for the rest.

Suppose we observe one particle of type $t_1$, three particles of type $t_2$ and three of type $t_3$, so the observation is $\{t_1, t_2, t_2, t_2, t_3, t_3, t_3\}$. Using steps 1-4 from the previous section, we obtain the following results. In Table 1 are given the estimate for the parameters of the grammar, and after summing up the respective probabilities, for the process we obtain that the nonzero terms in the offspring distribution are:

$$P(T_1 \to \{T_1, T_2\}) = 0.5, \quad P(T_1 \to \{T_2, T_3\}) = 0.5$$

$$P(T_2 \to \{T_1, T_3\}) = 0.5, \quad P(T_2 \to \{T_1, T_2\}) = 0.5$$

To reduce calculations, Monte Carlo samples are taken. Table 2 shows the results based on the average of three random samples of 20 permutations, and Table 3 for five samples of 10 permutations. It can be seen that the estimates for the parameters of the grammar obtained through these simulations slightly differ, but after summing up the respective terms, the estimates for the offspring probabilities of the MTBP are the same as with all permutations. Calculations are made in R (see [14]).

Table 2. Estimation for the parameters of the SCFG based on 3 samples of 20 permutations

| $T_1$: | $T_1$ | $T_2$ | $T_3$ | $T_1^T$ | $T_2^T$ | $T_2^T$ |
|---|---|---|---|---|---|---|
| $T_1$ | 0.0000 | 0.0654 | 0.0000 | 0.0000 | 0.2041 | 0.0000 |
| $T_2$ | 0.0535 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0654 |
| $T_3$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $T_1^T$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $T_2^T$ | 0.1770 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.1855 |
| $T_3^T$ | 0.0000 | 0.0656 | 0.0000 | 0.0000 | 0.1835 | 0.0000 |

| $T_2$: | $T_1$ | $T_2$ | $T_3$ | $T_1^T$ | $T_2^T$ | $T_2^T$ |
|---|---|---|---|---|---|---|
| $T_1$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.2448 |
| $T_2$ | 0.0000 | 0.0000 | 0.0000 | 0.2524 | 0.0000 | 0.0000 |
| $T_3$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $T_1^T$ | 0.0000 | 0.2476 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $T_2^T$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $T_3^T$ | 0.2552 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

**7. Conclusions.** In this work the connection between MTBP and SCFG was used in order to estimate the offspring probabilities of a multitype process.

Table 3. Estimation for the parameters of the SCFG based on 5 samples of 10 permutations

| $T_1$: | $T_1$ | $T_2$ | $T_3$ | $T_1^T$ | $T_2^T$ | $T_2^T$ |
|---|---|---|---|---|---|---|
| $T_1$ | 0.0000 | 0.0542 | 0.0000 | 0.0000 | 0.1331 | 0.0000 |
| $T_2$ | 0.0480 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0525 |
| $T_3$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $T_1^T$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $T_2^T$ | 0.2647 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.1454 |
| $T_3^T$ | 0.0000 | 0.0954 | 0.0000 | 0.0000 | 0.2068 | 0.0000 |

| $T_2$: | $T_1$ | $T_2$ | $T_3$ | $T_1^T$ | $T_2^T$ | $T_2^T$ |
|---|---|---|---|---|---|---|
| $T_1$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.1692 |
| $T_2$ | 0.0000 | 0.0000 | 0.0000 | 0.2903 | 0.0000 | 0.0000 |
| $T_3$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $T_1^T$ | 0.0000 | 0.2097 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $T_2^T$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $T_3^T$ | 0.3308 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

An approach was presented where a grammar corresponding to the process is constructed, and then a well-known EM algorithm for estimation of the parameters of the grammar is used. The results show that using such an algorithm it is possible to obtain the estimate in reasonable time. The Monte Carlo sampling approach also helps to reduce complexity.

## REFERENCES


[1] Asmussen S., H. Hering. Branching Processes. Birkhauser, Boston, 1983.

[2] Athreya K. B., P. E. Ney. Branching Processes, Springer, Berlin, 1972.

[3] Dempster, A. P., N. M. Laird, D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society,* **39** (1977), 1–38.

[4] Durbin R., S. R. Eddy, A. Krogh, G. Mitchison. Biological sequence analysis. Cambridge University Press, 1998.

[5] Geman S., M. Johnson. Probability and statistics in computational linguistics. a brief review. Mathematical foundations of speech and language

processing. (Eds M. Johnson, S. P.Khudanpur, M. Ostendorf, R. Rosenfeld), 2004.

[6] González M., J. Martín, R. Martínez, M. Mota Non-parametric Bayesian estimation for multitype branching processes through simulation-based methods. *Computational Statistics & Data Analysis,* **52** (2008), No 3, 1281–1291.

[7] Guttorp P. Statistical inference for branching processes. New York: Wiley, 1991.

[8] Harris T. E. Branching Processes. Springer, New York, 1963.

[9] Hyrien O. Pseudo-likelihood estimation for discretely observed multitype Bellman-Harris branching processes *Journal of Statistical Planning and Inference,* **137** (2007), No 4, 1375–1388.

[10] Kimmel M., D. E. Axelrod. Branching Processes in Biology. Springer, New York, 2002.

[11] Lari K., S. J. Young. The Estimation of Stochastic Context-Free Grammars Using the Inside-Outside Algorithm. *Computer Speech and Language,* **4** (1990), 35–36.

[12] McLachlan G. J., T. Krishnan. The EM Algorithm and Extensions. Wiley, 2008.

[13] Mode C. J. Multitype Branching Processes: Theory and Applications. Elsevier, New York, 1971.

[14] R Development Core Team (2010). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. `http://www.R-project.org`.

[15] Sankoff D. Branching Processes with Terminal Types: Application to Context-Free Grammars. *Journal of Applied Probability* **8** (1971), No 2, 233–240.

[16] Yakovlev A. Y., N. M. Yanev. Relative frequencies in multitype branching processes. *Ann. Appl. Probab.*, **19** (2009), No 1, 1–14.

[17] Yanev N. M. Statistical Inference for Branching Processes. In: Records and Branching Processes (Eds M. Ahsanullah, G. P. Yanev), Nova Sci. Publishers, Inc, 2008, 143–168.

*Nina Daskalova*
*Faculty of Mathematics and Informatics*
*St. Kliment Ohridski University of Sofia*
*5, J. Bourchier Blvd*
*1164 Sofia, Bulgaria,*
*e-mail:* `ninad@fmi.uni-sofia.bg`

*Received July 25, 2010*
*Final Accepted October 25, 2010*