Proceedings of Machine Learning Research 60:1–25, 2017 Conformal and Probabilistic Prediction and Applications

SCOT Approximation, Training and Asymptotic Inference

Mikhail Malyutov M.MALIOUTOV@NEU.EDU
Mathematics Department
Northeastern University
360 Huntington Avenue, Boston, MA 02115, USA
Paul Grosu PGROSU@GMAIL.COM
College of Computer and Information Science
Northeastern University
360 Huntington Avenue, Boston, MA 02115, USA

Editor: Alex Gammerman, Vladimir Vovk, Zhiyuan Luo, and Harris Papadopoulos

Abstract

Approximation of stationary strongly mixing processes by Stochastic Context Trees (SCOT) models and the Le Cam-Hajek-Ibragimov-Khasminsky locally minimax theory of statistical inference for them is outlined. SCOT is an m-Markov model with sparse memory structure. In our previous papers we proved SCOT equivalence to 1-MC with state space—alphabet consisting of the SCOT contexts. For the fixed alphabet size and growing sample size, the Local Asymptotic Normality is proved and applied for establishing asymptotically optimal inference. We outline what obstacles arise for a large SCOT alphabet size and not necessarily vast sample size. Training SCOT on a large string using clusters of computers and statistical applications are described.

Keywords: Strong mixing, strongly stationary sequences, Local Asymptotic Normality, Local Asymptotic Minimaxity, SCOT models, Edgeworth expansion.

1. Introduction

Strongly stationary sequences as an object of advanced Probability theory are studied in the *first part* of this paper culminating in their LAN property, section 5.

Approximability of strong mixing sequences by m-Markov Chain (m-MC) with large m belongs to the mathematical folklore and is widely used without rigorous definitions in the Information Theory, see Cover and Thomas (2006). In view of exponential complexity of general m-MC, ARMA-models were their most popular surrogates until **sparse memory** m-MC named VLMC was introduced in Rissanen (1983) for **compression aims**. We study novel conditions for sparse m-MC **approximation** of strong mixing stationary sequences called Stochastic Context Trees (SCOT) in section 2.2. Parameter m in SCOT depends in general on accuracy of the approximation and can be arbitrarily large, even infinite justifying appropriateness of a name alternative to VLMC .

The ergodicity of MC and Asymptotic Normality (AN) of additive state and transition functions of their paths has been subject of numerous studies starting from the pioneering works of A. A. Markov and S. N. Bernstein in the beginning of 20th century. Among many popular surveys—(Borovkov, 1998; Meyn and Tweedy, 1993; Tutubalin, 1992). Statistical inference for MC has become popular after (Billingsley, 1961; Roussas, 1972). The second of these references introduced the MC Local Asymptotic Normality (LAN) following the Le Cam-Hajek asymptotic locally minimax inference theory. An elementary exposition of this theory is in Tutubalin (1992); Veretennikov (2000). The *traditional asymptotics* with *fixed MC* and growing sample size N is considered in all these references. Our section 5 outlines the simpler straightforward LAN derivation for finite MC with fixed alphabet following (with substantial revisions) Tutubalin (1992); Veretennikov (2000). The key point is a new Asymptotic Normality proof for additive **transition** functions which replaces the popular much longer way (for more elementary case of additive **state** functions) based on a reduction to the more general Martingale Central Limit Theorem. The latter approach involves a cumbersome Poisson-like inverse problem solution which is not straightforward (Meyn and Tweedy, 1993).

Our statistical SCOT modeling (Ryabko et al., 2016; Malyutov et al., 2013) of financial data discovered a small size of their context tree, while literary texts showed the number of SCOT contexts $m > 2000$; m is the size of 1-MC alphabet equivalent to SCOT, if the corresponding SCOT has a perfect memory, (Zhang, 2016), in (Ryabko et al., 2016; Malyutov and Zhang, 2015) another name: 'TailClosed' is used for the same object. Otherwise, a perfect memory envelope of the original SCOT, (Zhang, 2016) should be used with even larger alphabet size.

The literary texts examples in Ryabko et al. (2016); Malyutov et al. (2013) show that the traditional asymptotics for deriving AN of additive SCOT functions can be inadequate. A similar change of asymptotics has been suggested by A. N. Kolmogorov for statistical classification of objects characterized by many normally distributed characters each, see Deyev (1970), where Kolmogorov's ideas are exposed.

To illustrate what happens when both m and sample size N grow simultaneously, we consider in subsection 3.1 the spectral decomposition of cyclic random walks nicely exposed in Feller (1967), (pp. 377–378 and 434–435).

Our study of alternative asymptotics uses the first order Edgeworth expansion for the additive functions — see Bolthausen (1980, 1982); Malinovsky (1987); Jensen (1989) among many publications. The principal multiplier $(\mu_3/\sigma^3)/\sqrt{N}$ of the residual term may grow with m which worsens the precision of approximation, see section 4. Here μ_3, σ are the stationary third moment and standard deviation of X_i respectively. Our discussion of alternative asymptotics is novel.

Sections 6-7 constitute the *mathematical statistics part* of this paper. We outline asymptotically optimal inference in estimation and testing local hypotheses under LAN validity.

Section 7 justifies SCOT homogeneity testing results of Ryabko et al. (2016); Malyutov et al. (2013) in the framework of local (contiguous) alternatives theory under LAN. Testing very distant alternatives was exposed earlier in Ryabko et al. (2016); Malyutov et al. (2013) for an example of screening out active inputs of a multivariate regression model with colored noise using the large deviations probability results. Results of section 7 justify the effectiveness of the homogeneity nonparametric testing in section 8.4. Estimation of transition probabilities in sliding windows is aimed at distinguishing abrupt changes in SCOT model from its small deviations.

The speed up of SCOT training of the sparse m-MC with a large alphabet size or for multichannel online SCOT training prompted development in section 8 of our novel

parallel implementation of the algorithms developed earlier in Rissanen (1983); Mächler and Bühlmann (2004) et al without quoting their consistency results.

Our simulation algorithm for section 3.1 is outlined in Appendix.

2. Approximation by SCOT

2.1. Review of previous approaches to m-MC approximations

We consider a strictly stationary process $X(t)$ over a finite alphabet $\mathcal{X}$ of cardinality $|\mathcal{X}|$ and discrete time: $X_t, -\infty < t < \infty$, with potentially infinite memory which can be approximated uniformly by an m-Markov chain (UA-m-MC condition).

By this we mean that for any $\varepsilon > 0$ there exists a positive integer $m(\varepsilon) > 0$ such that

$$|P(X_0|X_{-\infty}^{-1}) - P(X_0|X_{-m}^{-1})| < \varepsilon$$

for almost every X_0 and past sequences $X_{-\infty}^{-1}$.

Remark 1. *Apparently, a uniform version of absolute mixing — attributed to A.N. Kolmogorov in Volkonskii and Rozanov (1959) — with exponential memory decay can guarantee a uniform approximation by an m-MC.*

Numerous conditions of strong mixing sequences are reviewed in Bradley (2005). Theorem 1.2 and Remark 3 in Bradley (1989) assure that a very artificial stronger version of absolute regularity is equivalent to the fact that the sequence is m-MC for some m.

2.2. Sparse m-MC approximations

Assume now the UA-m-MC condition of a stationary sequence $x_{-\infty}^{\infty}$ and fix $m(\varepsilon)$ of the approximating m-MC which is assumed ergodic. Sequences x_1^m are called m-grams. Let us introduce *contexts* for each of $|\mathcal{X}|^m$ different realizations a_1^m of m-gram.

The context to m-gram a_1^m is its final part of minimal length $l(a_1^m)$ such that the conditional distributions $P(X_{m+1}|x, a_{m-l}^m)$ do not depend on prefix x up to joint error probability $< \varepsilon$. This statement is described by simultaneous validity of obvious $A \times (A-1)$ double inequalities. Not occurring m-grams are ignored. To streamline introduction, we assume that there are no such m-grams. Such a twice approximated stationary sequence will be called ε-SCOT with small abuse of notation.

Finally, the *memory spectrum* $\mathcal{M} = m_1^{2^m}$ is the $|\mathcal{X}|^m$-vector of context lengths along $|\mathcal{X}|^m$ paths from the root to the past showing distribution of the memory size sufficient for predicting the root symbol distribution.

We combine all preceding developments into the following:

Definition. If a UA-m-MC has a memory spectrum $\mathcal{M}$, then $X_{-\infty}^{\infty}$ is an ε-SCOT with the corresponding context length distribution.

We say that $X_{-\infty}^{\infty}$ has a *sparse m-MC representation*, if the average context length satisfies:

$$|\mathcal{X}|^{-m} \sum_{i=1}^{|\mathcal{X}|^m} Em_i << m.$$

Widespread sparse processes in nature phenomenon are explained by the 'Occam razor' or 'Bottleneck' popular philosophical principles.

Averaging context lengths over their stationary distribution gives a better sparsity indicator—the mean prediction length. The average prediction length is a preliminary indicator before the stationary distribution for the contexts is found. The median, or another quantile collection, or other functions of $\mathcal{M}$ over the stationary distribution can be also used for defining sparsity. Many examples of stationary distribution evaluation among various SCOT modeling results are in Ryabko et al. (2016).

If under UA-m-MC condition, we are given a long string with a vast collection of m-grams, then the probability context length distributions can be replaced with their corresponding consistent frequencies. This allows the sparse m-MC *consistent training* based on choosing as contexts the strings of the past with small Empirical Shannon Information (ESI), see Galves and Loecherbach (2008); Mächler and Bühlmann (2004); Rissanen (1983). A version of SCOT training on a cluster of computers valid for a large alphabet is described in our section 8.

Remark 2. Another indicator of sparsity is the entropy rate of n-string which is much lower for the UA-m-MC with sparse memory than a general linear one. For example, the entropy rate of arbitrary n-string in the Comb model, (Ryabko et al., 2016), does not exceed a const.

3. Asymptotic Normality of additive transition functions

Ryabko et al. (2016); Malyutov and Zhang (2015) and especially Zhang (2016) established the equivalence of a perfect memory sparse SCOT to 1-MC with state space consisting of the m-MC contexts which we call alphabet $\mathcal{A}$ of cardinality A . For not perfect memory sparse SCOT, its perfect memory sparse *envelope* studied in (Zhang, 2016) plays this role. Thus, by first applying UA-m-MC and $\mathcal{M}$ conditions, we reduced a stationary sequence to an m-MC with sparse memory structure, and now **reduce it further to a 1-MC with alphabet $\mathcal{A}$**.

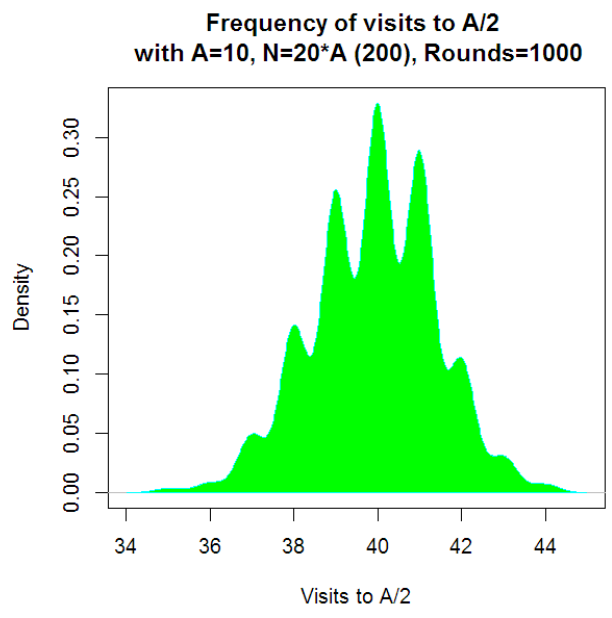
We develop further asymptotic theory mostly for fixed A and large sample size and, therefore, for a finite ergodic MC suppressing arbitrary $\varepsilon > 0$ in previous approximations.

Let $X_i, i = 0, 1, \dots$ be the subsequent values of ergodic MC with alphabet $\mathcal{A} = \{1, \dots, A\}$ and transition matrix $P = (p_{jk}, 1 \leq j, k \leq A); \pi = \pi(0)$ be the row-vector-initial distribution of X_0 and π^* be the stationary π. Finally, let $f(\cdot, \cdot)$ be a real function on $\mathcal{A} \times \mathcal{A}$. We call $S_n = \sum_{i=0}^{n} f(X_i, X_{i+1})$ an *additive transition function* (ATF) of MC X_i. An important ATF example is the *loglikelihood* $l_n(\theta)$ of a string X_0^n depending on vector θ of all transition probabilities and the *loglikelihood ratio* $r_n(\theta, \theta')$ which asymptotic representation in section 5 establishes the LAN property.

3.1. Ouverture: asymmetric cyclic RW example

To illustrate what happens when both A and sample size N grow to infinity, let us consider the asymmetric cyclic random walk (RW).

The alphabet consists of equidistant circumference points $\exp(2ik\pi/A), k = 0, 1, \dots, A-1$, i is the imaginary unit. The asymmetric cyclic random walk stays in the same state or jumps to the nearest left state with probabilities 1/2. Introduce $\theta = \exp(2i\pi/A), s_r =$



Histogram

$((1/2)(1+\theta^r)$. Equation (2.11) of (Feller, 1967) establishes the power n of transition matrix spectral decomposition

$$p_{jk}^{(n)} = (A^{-1}) \sum_{r=0}^{A-1} \theta^{r(j-k)} s_r^n. \tag{1}$$

We see from (1) that eigenvalues of the transition matrix are $O(A^{-1})$ apart as $A \to \infty$ which means that we cannot separate the maximal of them from the rest and restrict spectral expansion to just one 'maximal summand'. The term $p_{jk=0}^{(A/2)}$ corresponds to the additive state function for the indicator function of state $A/2$. Obviously, this function takes the value 0, if the initial state is 1 and number of summands less than $A/2$. Distribution of the sum is far from Normal, if few more summands are involved.

This fact is displayed in empirical histogram of visits to the state $A/2$ (fig.1), where the sample size N is 20 times more than A as a result of 1000 simulations. It shows several slightly intersected clusters far from the overall Normal histogram.

3.2. The AN of additive MC functions under fixed alphabet size

The most popular derivation of the CLT for MC nowadays is based on a reduction to the more general Martingale CLT which requires rather cumbersome approximations to the Poisson inverse-problem-like solution which is not straightforward — see e.g. Meyn and Tweedy (1993).

To compose parts of derivation together, we outline a simpler approach based on results of the Russian Probability school.

Let $\mathbf{e}$ be A-column consisting of ones. For a real number t, introduce a new matrix $P(t)$ with entries $p_{jk}(t) = p_{jk}\exp(tf(j,k))$ and start with an elegant expression of S_n' moment generating function (MGF):

$$F_n(t) = E_\pi \exp(tS_n) = \pi P^n(t)\mathbf{e}. \tag{2}$$

The proof of insufficient for our aims particular case of additive state function (ASF) (where $f(\cdot,\cdot)$ depends only on its second argument) is displayed in (Tutubalin, 1992), pp. 230-232, and erroneously attributed there to A. A. Markov. The origin of this formula remains unclear to us. A. A. Markov actually used a cumbersome method of moments for deducing AN of S_n. We omit the detailed derivation of this formula. It is straightforward via sequential conditioning. At first $E(E(F_t|X_0^{n-1})) = F_{n-1}(\cdot)P(t)e$, then similar conditioning on X_0^{n-2}, etc.

To simplify further exposition, let us assume that all entries of $P = P(0)$ (and therefore also of $P(t)$) are positive. In view of ergodicity of $P = P(0)$, this is certainly valid for some power of $P = P(0)$ — see Feller (1967) — which is sufficient for our aims in this paper. Thus, $p_{jk}(t) = p_{jk}\exp(tf(j,k))$ is also strictly positive. The Perron-Frobenius theorem implies that the isolated eigenvalue $\lambda(t)$ of $P(t), 0 \leq t < \infty$ with the largest real part exists. Due to analicity of $P(t)$ and the theorem of implicit functions, this unique root $\lambda(t)$ of the equation

$$det(P(t) - \lambda I) = 0, \tag{3}$$

is an infinitely differentiable function of t in a neighborhood of $\lambda(0) = 1$. Attached to eigenvalue $\lambda(t)$ are row eigenvector $q_t \to \pi^*$ and column eigenvector $e(t) \to e$ as $t \to 0$ infinitely smoothly depending on t, with unit scalar product. Then $P_1(t) = \lambda(t)e(t)q_t$ is such that $P(t) - P_1(t) = P_2(t)$ is exponentially smaller than $P_1(t)$ due to the Perron-Frobenius theorem. For our aims in this paper, $P_2(t)$ can be ignored.

Ergodicity: A normalized additive MC functions (ATF and ASF) shifted with time are obviously a stationary process converging to $\mu = ES_n/n$ as $n \to \infty$.

The proof — see Tutubalin 1992 (pp. 236–237) — of the AN of normalized additive ASF functions via applying twice the L'hospital's rule to its MGF is pretty standard given our representation of its MGF and similar to that in the IID case, see e.g. Grinstead and Snell (2006). The proof in the ATF case is essentially the same.

To prove the weak convergence to the limiting Normal approximation (possibly singular) under usual $\sqrt{N}$ normalization for **centered** ATF, we evaluate the second derivative of its 'reduced' MGF $P_1^0(t)$ at $t = 0$.

Terms involving the first derivative of the centered $P_1^0(t)$ vanish at $t = 0$ due to centering. Only one term remains

$$[(\pi(t/\sqrt{N})e)(\pi^* e(t/\sqrt{N})) + o(1)][1 - (t\sigma)^2/2n]^n \to \exp -(t\sigma)^2$$

as $n \to \infty$. This finishes the proof according to well-known classical Probability approximation theorems since the limiting MGF is that of the standard Normal distribution.

Of some interest is that the limiting distribution under standard normalization can be singular due to the null limiting variance.

As a consequence, in this case there is no need for $\sqrt{N}$ normalization, and the residual distribution is bounded.

A simple example of such anomaly for additive state function is the *symmetric cyclic RW* with four states and equally likely transitions to each of two neighbors, and alternating ± 1 function between neighboring states. Values ± 1 necessarily alternate also in time killing each other. Thus $S_{2n} = 0$, while $S_{2n+1} = \pm 1$ for all n and the standard $1/\sqrt{n}$ normalization provides the limiting null variance.

4. The Edgeworth expansion of additive MC functions under fixed alphabet size

Asymptotic expansion of additive functions appeared in Bolthausen (1980, 1982), Malinovsky (1987), Jensen (1989) under various conditions which certainly include the case of a duly smoothed finite ergodic MC. An appropriate smoothing procedure is described in theorem 2, XVI,4, (Feller, 1970).

In Malinovsky (1987) the *first terms* of asymptotic expansion under Cramer-type conditions are:

$$P((N^{-1/2}\sum_{i=1}^{N} f(x_i)) \le x) = \Phi_\sigma(x) + \phi_\sigma(x)q(x)N^{-1/2} + O(N^{-1}).$$

Here ϕ and Φ are PDF and CDF of the central Normal RV with StD σ, q is expressed in terms of the first Hermite polynomial $1 - x^2$.

The principal multiplier $(\mu_3/\sigma^3)/\sqrt{N}$ of the residual term may grow with A which worsens the precision of approximation. Here μ_3, σ are the stationary central third moment and standard deviation of X_i respectively.

In particular, for our circular MC of section 3.1 and the indicator function of state $A/2$ as an example of additive function, the mean is $A^{-1}, \mu_3 = A^{-1} + O(A^{-2}), \sigma^2 = A^{-1} + O(A^{-2})$. Thus, the principal multiplier of the residual $O(A^{1/2}/\sqrt{N} \to 0$, only if $(A/N) \to 0$ as $N \to \infty$.

5. The Local Asymptotic Normality of SCOT under a fixed context cardinality

One of our principal aims is to outline the *Local Asymptotic Minimaxity* (LAM) and the Locally Asymptotically Most Power (LAMP) of the likelihood based inference and of its certain approximations. The LAM in parameter estimation as formally defined further in section 6, means that the deviation of the estimate from the true parameter value θ^* is as minimal as possible in the local minimax sense.

It is implied by the LAN condition — see e.g. Veretennikov (2000); Roussas (1972) — for MC which will be introduced immediately. The principal role in the LAN proof is played by the AN of the ATF functions — established in section 4 and missing in Veretennikov (2000). Two elementary corollaries of ergodicity ending the LAN proof and best exposed in Veretennikov (2000) are omitted here.

The Local Asymptotic Normality (or simply LAN) introduced in Le Cam (1960) is the following decomposition of

$$r_n(\mathbf{u}) = \ln[P_{\theta+n^{-1/2}\mathbf{u}}((X_0^n))/P_\theta((X_0^n))], \mathbf{u} \in \mathbf{R}^A :$$

$$r_n(\mathbf{u}) = u^T \lambda - (1/2)\mathbf{u}^T J \mathbf{u} + \psi_n(\mathbf{u}),$$

where

$$\lambda \sim N(0, J), J = E_{\pi^*} \partial r(\theta) \partial r(\theta)^T$$

is the limit of the mean in the ATF $r(\cdot)$ Jacobian multivariate AN approximation.

and $\psi_n(\mathbf{u})$ converges in $P_\theta(X_0^n)$- probability to zero.

This expansion for a univariate parameter via the Taylor expansion of the second order is proved in Veretennikov (2000) referring to the much more involved exposition in (Roussas, 1972) for the AN proof of the ATF in general case under standard regularity conditions.

The uniformity of the residual $\psi_n(\mathbf{u})$ convergence in $P_\theta^{(n)}$- probability to zero can be proved by the more elegant Lagrange-type integral representation of the second order residual in the Taylor expansion as in Malyutov and Protassov (2002). Namely, for all $K > 0, a > 0$

$$\lim_{n \to \infty} P_{\theta + n^{-1/2}\mathbf{u}, \sup ||\mathbf{u}|| < K}(|\psi_n(\mathbf{u})| > a]) = 0.$$

6. The Local Asymptotic Minimaxity of Likelihood-Ratio-like tests under a fixed alphabet size

Let the distribution family P_θ satisfy LAN condition in $\theta = \theta^*$ with the identity Fisher information matrix, $|| \cdot ||$ be the Euclidean norm. A function $w(\cdot) : \mathbf{R}^p \to \mathbf{R}^+$ is called bowl-shaped if $\{\mathbf{u}| \; w(\mathbf{u}) \le a\}$ are closed bounded symmetric convex sets for any $a \ge 0$. An increasing continuous bowl-shaped function $w(\cdot) : \mathbf{R}^+ \to \mathbf{R}^+, w(0) = 0$, is called a loss function.

The fundamental Hajek's lower bound for the LAM-risk of any estimate T_n for any loss function $w(\cdot)$ and $\delta > 0$:

$$\liminf_{n \to \infty} \sup_{||\theta - \theta^*|| < \delta} E_\theta w(n^{1/2} ||T^n - \theta||) \ge \int w(\mathbf{u})(2\pi)^{-1/2} e^{-|\mathbf{u}|^2/2} d\mathbf{u},$$

holds. In general, the positively definite Fisher information J determines the norm in the risk function definition.

The LAM property of the Maximum Likelihood (ML) estimate and of the Fisher score update to ML given a qualified consistent prior estimate for θ are exposed in Veretennikov (2000); Roussas (1972). Malyutov and Protassov (2002) shows sufficiency of a usual consistent estimate for θ for LAM of the Fisher score update given the uniform LAN property.

The third Le Cam's lemma (Chibisov, 2009; Roussas, 1972) proves that the AN of a statistic under the null hypothesis implies its AN under the alternative distribution provided contiguity — defined in Roussas (1972) — and the LAN condition.

7. Locally asymptotically optimal tests

The most transparent overview of the Locally Asymptotically Most Powerful (LAMP) tests under LAN condition for I.I.D samples is in Chibisov (2009). Given LAN property, it differs insignificantly from the one for MC in Roussas (1972).

The main distinction of the LAMP approach originated in Le Cam's works from the traditional one, is that the 'close' alternatives $\mathbf{u}(n^{-1/2})$ are considered for the sample size $n \to \infty$. This enables limiting positive significance level and power asymptotically and a transparent application of the familiar testing shift theory for multivariate Normal. We now give schematic simplified overview of this theory following Chibisov (2009) and shortening our notation for transparency in an obvious way.

The Neyman-Pearson lemma gives the most powerful test of significance level α against alternative $\mathbf{u}(n^{-1/2})$ as

$$r_n(\mathbf{u}) = \ln[P_{\theta+n^{-1/2}\mathbf{u}}((X_0^n))/P_\theta((X_0^n))] > C_{n,\alpha},$$

with parameter $C_{n,\alpha}$ determined from equation $P_{n,0}(r_n > C_{n,\alpha}) = \alpha$.

The LAN condition converts this into the asymptotic equality $C_{n,\alpha} = z_\alpha\sqrt{J}\mathbf{u} - \mathbf{u}^T J\mathbf{u}/2$ which is equivalent to

$P_{n,0}(r_n < x) \to \Phi((x + \mathbf{u}^T J\mathbf{u}/2)/\sqrt{\mathbf{u}^T J\mathbf{u}})$

The power is $\beta_{n,\mathbf{u}} = P_{n,\mathbf{u}}(r_{n,\mathbf{u}} > C_{n,\alpha})$ as $n \to \infty$ implying

$P_{n,\mathbf{u}}(r_n < x) \to \Phi((x - \mathbf{u}^T J\mathbf{u}/2)/\sqrt{\mathbf{u}^T J\mathbf{u}})$

Thus, $\beta_{n,\mathbf{u}} = P_{n,\mathbf{u}}(r_n > z_\alpha\sqrt{J}\mathbf{u}) \to 1 - \Phi(z_\alpha - \sqrt{J}\mathbf{u}) = \Phi(\sqrt{J}\mathbf{u} - z_\alpha)$ which means — see e.g. Chibisov 2009, (§8.1.19) — that the limiting asymptotic power of our test is asymptotically maximal for every given alternative $\mathbf{u}$ in view of the Neyman-Pearson lemma. Thus, our test is LAMP.

Let us apply the preceding theory to the homogeneity of multivariate distributions of the large strongly stationary ergodic training string T and a query string Q. We use the nonparametric test of Malyutov et al. (2013).

The first stage is estimation of the SCOT model of the string T following the algorithm in Mächler and Bühlmann (2004). We refer to this publications for the details.

We assume

1. the T's and Q's good approximability by a sparse SCOT and
2. fulfillment of the LAN condition for the equivalent 1-MC over their contexts.

We cut the query string into K slices of the same length. Then, using the SCOT model of T we find the loglikelihoods $L_Q(k)$ of query slices Q_k and of strings S_k simulated from the training distribution of the same size as $Q_k, k = 1, \ldots, K$, for constructing simulated strings, see e.g. algorithm in Mächler and Bühlmann (2004).

We then find log-likelihoods $L_Q(k)$ of Q_k, $L_S(k)$ of S_k using the derived probability model of the training string and the average $\bar{D}$ of their difference D which approximates the likelihood ratio statistic discussed above. The averaging over slices is used for empirical evaluation of the loglikelihood variances since our testing homogeneity problem is completely nonparametric.

We assume though that the multivariate distributions of the training and the query strings are contiguous. In particular, for literary applications this assumption means that both texts are written in the same language, and admissibility of texts is the same for T and Q.

Next, due to the asymptotic normality of log-likelihood increments both for the null hypothesis and alternative (third LeCam's lemma), we can compute the usual empirical variance V of $\bar{D}$ and the t-statistic t as the ratio $\bar{D}/\sqrt{V}$ with $K-1$ degrees of freedom

(DF). We find K^* from the empirical condition that $t(K^*)$ is maximal. Then, the p-value of homogeneity is evaluated for the t-distribution with $K^* - 1$ DF.

8. Algorithms

The first SCOT training algorithm was constructed and its consistency proved in Rissanen (1983). The open source algorithm 'Context' in language R appeared in Mächler and Bühlmann (2004). Our statistical application of this algorithm to stationary or piecewise stationary financial, literary and seismic data were described in Malyutov et al. (2013); Ryabko et al. (2016). Refined consistency proofs of the 'Context' consistency are e.g. in Bühlmann and Wyner; Galves and Loecherbach (2008). This section describes a novel parallel implementation of the algorithm similar to 'Context' which is created for fast processing more complex data sets including those with larger alphabet sizes. We do not review proofs of 'Context' consistency.

Algorithm 'Context' in Mächler and Bühlmann (2004) is reasonably fast on a PC, if sparsity of the MC memory structure is valid and the alphabet size A is not larger than 27. The criterion of selecting contexts according to their appropriately small Empirical Shannon Information (ESI), (see p.13, section 8.2) usually stops back-processing of the training string long before the chosen horizon. All directions backwards from the root are processed in parallel making the algorithm much faster.

8.1. *Pre-processing. Alphabet Correction of Corporas*

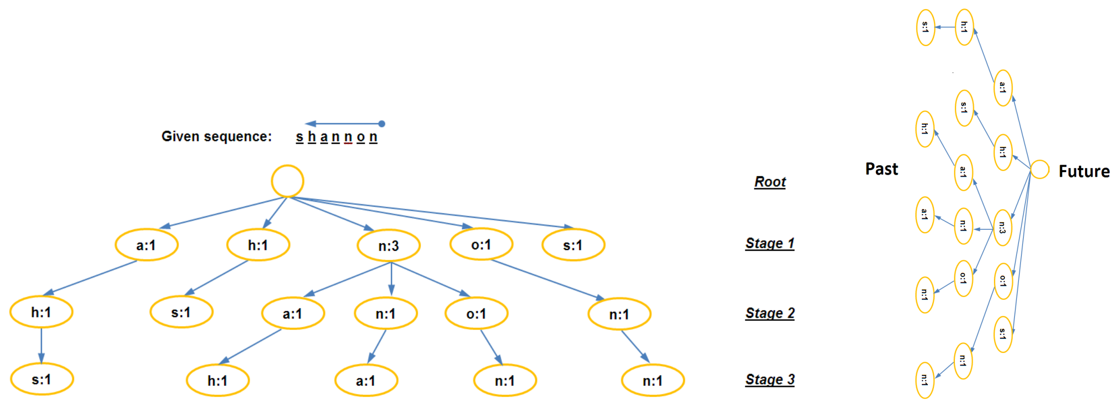
The process of transforming a literary corpus for converting all characters to lower-case, validating against an alphabet and replacing non-alphabet symbols to spaces or empty-spaces as appropriate[1]. Any characters that are in the alphabet will be kept, and any multi-space sequences are converted to one space. The only exception is with apostrophes, which are converted to an empty space and newline character which is converted to a space in order to have the text be a sequential sequence of alphabet-validated corpus. All names and dialogues also need to be removed from the corpora in order for proper analysis.

8.2. *Training Stochastic COntext Trees (SCOT)*

The *Stochastic COntext Trees (SCOT)* training program — written using the Python programming language — builds the stochastic trees starting from stage 1 and proceeding to the horizon stage of interest. Potential contexts having an ESI value smaller than ϵ become contexts, and would be omitted from processing in the following stages. Another improvement in parallelism is processing of a potential context by hashing into sets — for storing count statistics — and for quickly determining if it should be processed on one node of many, by modulo of the hash with the total number of compute nodes. The assumption here is that there we have many (hundreds) of compute nodes available to process a corpus into a SCOT, and we follow the MapReduce (Dean and Ghemawat, 2004) paradigm in building up a SCOT. The allocation of compute nodes was performed using the open-source SLURM package (SLU), developed at the Lawrence Livermore National Laboratory. The

1. For example, a dash (-) would become a space (" "), while a period (.) will become an empty space("").

function that initiates the construction of a SCOT is the *run_stage*() function described on the following page. The key to the determination of a context if ESI falls below the ϵ threshold.



(a) When parsing a corpus one can encode sequences of *string-contexts* with their corresponding counts, by traversing to the past.

(b) Contexts can be joined as a tree.

Figure 1: A SCOT being trained for a given sequence.

The leaves of a SCOT will contain the probabilities at the root (future) symbol:

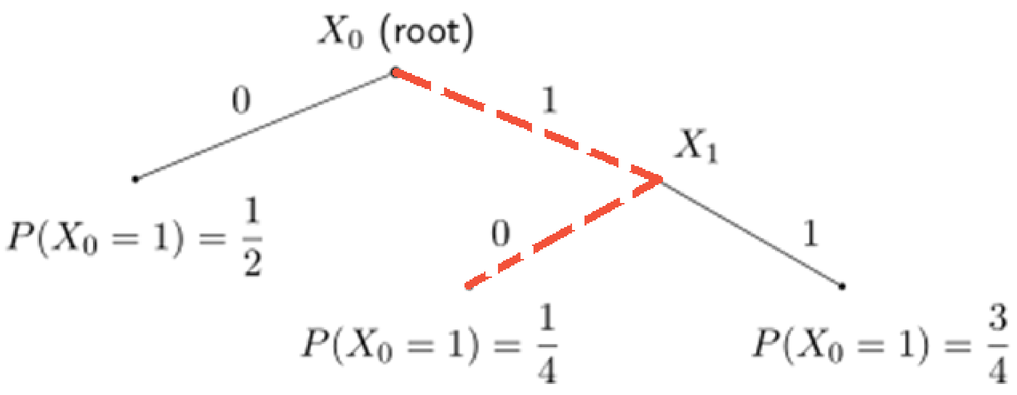

X_0 (root)
0
1
X_1
0
1
$P(X_0 = 1) = \frac{1}{2}$
$P(X_0 = 1) = \frac{1}{4}$
$P(X_0 = 1) = \frac{3}{4}$

Figure 2: *EXAMPLE*: Notice that in the sequence 001$\underline{01}$**1**1101$\underline{01}$**0**11 - for the two underlined contexts - $P(\ 1 \mid 01\ ) = \frac{1}{4}$, and the complement would be $P(\ 0 \mid 01\ ) = 1 - \frac{1}{4} = \frac{3}{4}$. *The probability of a future symbol given its past is* $P(\ Symbol_{FUTURE} \mid Context_{PAST}\ )$.

As all stages are performed using the same (*run_stage*()) algorithm, and start from 1. If for a specific stage some subsequences become contexts because they fall below the ϵ threshold, then those will not be processed in subsequent stages.

In the ($run\_stage()$) algorithm — for each stage — the first input is the full sequence of the text ($full\_seq$), which is used for building up the counts for the matrix (Fig. 3). The second input is the stage to be processed ($Stage$). The third input is the set of confirmed contexts ($Confirmed\_Contexts$) of previous stages, which if encountered in the current stage can be ignored from processing. The fourth input is the (ϵ) threshold used by the $RissanenESI()$ in (Algorithm 3). The fifth and sixth inputs ($node\_id$ and $total\_nodes$) are used by (Algorithm 6) in determining if a possible subsequence for context-validation should be processed on the current node for optimal distributed computing. The subsequences that are equal to or fall below the ϵ threshold are considered contexts, and will be saved into a file called `contexts_stage_NUMBER.csv` — with the context string appended to a file called `contexts_list.csv` for use in subsequent stages — otherwise they are saved into a file called `esi_stage_NUMBER.csv`. The format for each saved row in either staged files is as follows:

CONTEXT , ESI Score , Alphabet probability distribution at the leaf

The leaf probability distribution is formatted as a quoted alphabet symbol, with its determined probability value. Such a format allows for direct selection of the context or ESI value — since it is comma-separated — and for the symbol probability as it is subsequently semicolon-separated. Below is an example of one such row:

```
ann, 1.89413252, " "=0.03653586 ; "a"=0.03653586 ; "b"=0.03653586 ;
"c"=0.03653586 ; "d"=0.03653586 ; "e"=0.03653586 ; "f"=0.03653586 ;
"g"=0.03653586 ; "h"=0.03653586 ; "i"=0.03653586 ; "j"=0.03653586 ;
"k"=0.03653586 ; "l"=0.03653586 ; "m"=0.03653586 ; "n"=0.03653586 ;
"o"=0.05006766 ; "p"=0.03653586 ; "q"=0.03653586 ; "r"=0.03653586 ;
"s"=0.03653586 ; "t"=0.03653586 ; "u"=0.03653586 ; "v"=0.03653586 ;
"w"=0.03653586 ; "x"=0.03653586 ; "y"=0.03653586 ; "z"=0.03653586
```

On the following page is the algorithm for processing a stage:

Algorithm 1: The *run_stage*() function on one node.

Inputs:
full_seq: The sequence of symbols in the corpus.
Stage: The current stage being processed.
Confirmed_Contexts: The list of confirmed contexts, which encountered can be ignored from being processed for context-checking.
ϵ: The threshold value for ESI.
node_id: Node ID out all nodes.
total_nodes: Total number of compute nodes.

Outputs:
Writes the contexts ($\leq \epsilon$) and non-contexts ($> \epsilon$) for a specific stage.

```
1  begin
2      // Have a baseline minimum ESI value to compare with
       $previously\_a\_context\_value = -10^7$
3      // Filter out any text that is already an established context, and
4      // Process the counts for new non-contexts
       $populate\_NonContextSequenceCounts\_and$
5        $\_FilterOutEstablishedContexts\_in\_Corpus($
6        $full\_seq, Stage + 2, Stage, 0, |full\_seq| - 1,$
7        $Confirmed\_Contexts, node\_id, total\_nodes)$
8      // Initialize the current node's ESI statistics for all sub-sequences
9      // processed $stage\_ESI = \{\}$
10     // Process all sequences from all buckets, distributed by node-id
11     for $SEQUENCE\_BUCKET \in SEQUENCE\_BUCKET\_LIST$ do
12         for $SEQUENCE \in$
           $SEQUENCE\_BUCKET\_LIST[SEQUENCE\_BUCKET]$ do
13             if should_context_be_processed_on_this_node(
               SEQUENCE_TO_PROCESS, node_id, total_nodes ) then
14                 $SEQUENCE\_ESI =$
                   $RissanenESI(SEQUENCE, alphabet)$
15                 $stage\_ESI[SEQUENCE] = SEQUENCE\_ESI$
```

The *run_stage*() function is continued on the following page.

Algorithm 2: (Continued) The *run_stage*() function on one node.

```
1  begin
2      // Build Contests if ESI $\leq \epsilon$
3      $contexts\_less\_equal\_epsilon = \{\}$
4      for $Context \in stage\_ESI$ do
5          // The 0-position stores the ESI value
6          if $stage\_ESI[Context][0] \leq \epsilon$ then
7              if $stage\_ESI[Context][0] > previously\_a\_context\_value$ then
8                  $contexts\_less\_equal\_epsilon[Context] = stage\_ESI[Context]$
9      // Build Non-Contests if ESI $> \epsilon$
10     $ESI\_stage\_with\_no\_contexts = \{\}$
11     for $Context \in stage\_ESI$ do
12         if $stage\_ESI[Context][0] > \epsilon$ then
13             $ESI\_stage\_with\_no\_contexts[Context] = stage\_ESI[Context]$
14     // Write only greater than epsilon
15     $write\_not\_contexts\_file(ESI\_stage\_with\_no\_contexts, Stage, node\_id)$
16     // Write out Contexts in another file
17     $write\_contexts\_file(contexts\_less\_equal\_epsilon, Stage, node\_id)$
18     for $Context \in contexts\_less\_equal\_epsilon$ do
19         $confirmed\_contexts\_list =$
           $update\_confirmed\_contexts\_list(new\_context)$
20     $write\_contexts\_list(contexts\_less\_equal\_epsilon, node\_id)$
```

The key to the determination of a context if ESI falls below the ϵ threshold:

$$ESI = \sum_i \sum_j i.s.j * log_2 \left(\frac{i.s.j}{\frac{i.s}{\frac{s.j}{s}}} \right) \leq \epsilon$$

This is performed by populating a matrix for each context-check, which is initially populated by the counts of each substring occurrence, and Laplace-smoothened by adding 0.1 to the matrix, which is described in [Fig. 3].

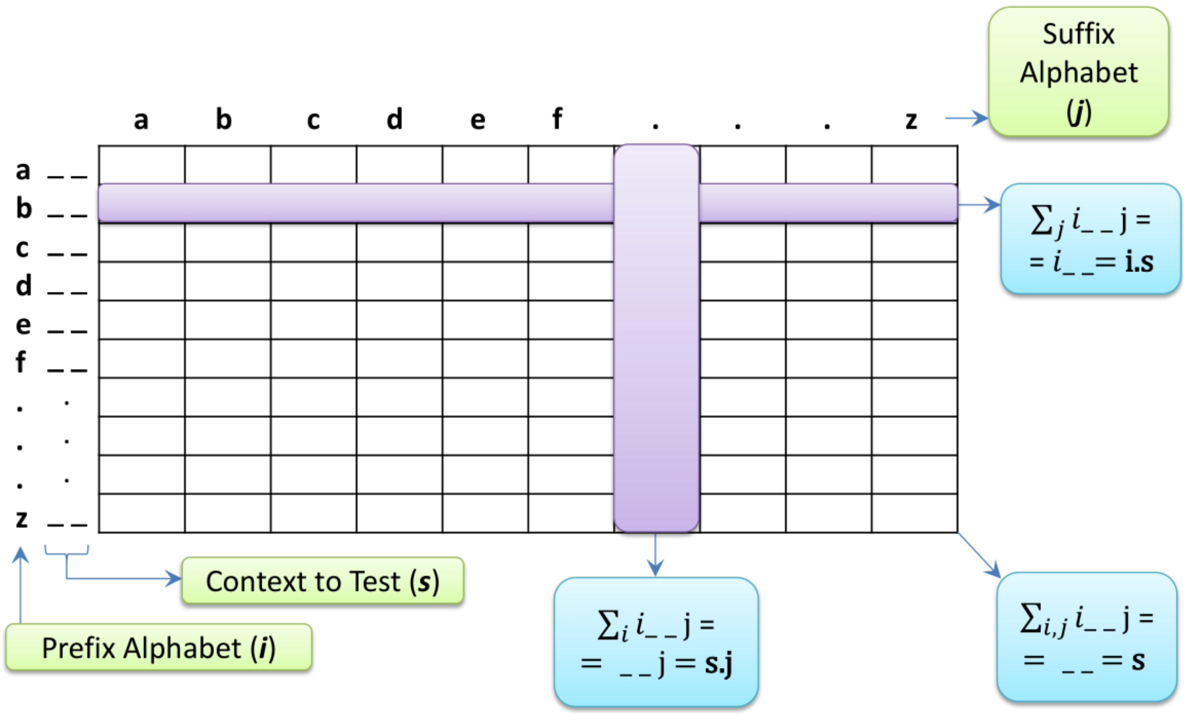


Figure 3: The matrix for used for ESI calculation regarding context-checking.

The function performing this calculation is called *RissanenESI*() and is described in the following pages.

Algorithm 3: The *RissanenESI*() function on one node.

Inputs:
current_context: The context to check and build the matrix for.
alphabet: The alphabet to use.

Outputs:
Returns a tuple of the ESI value and the leaf-distribution.

```
1  begin
2      // The Laplace smoothing parameter
3      $laplace\_offset = 0.1$
4      // Initialize the matrix with zeros
5      $count\_matrix = zeros(|alphabet|, |alphabet|)$
6      // Populate the matrix
7      for $prefix\_char\_index \in \{0, \ldots, |alphabet| - 1\}$ do
8          for $suffix\_char\_index \in \{0, \ldots, |alphabet| - 1\}$ do
9              $lookup\_sequence = alphabet[prefix\_char\_index] + current\_context + alphabet[suffix\_char\_index]$
10             // Get the sequence counts previously populated
               $count\_matrix[prefix\_char\_index][suffix\_char\_index] = retrieve\_STAGE\_SEQUENCE\_COUNTS(lookup\_sequence)$
11     $i\_symbols\_j\_count\_matrix = count\_matrix + laplace\_offset$
12     $s = 0.0$
13     // Build i.s
14     $i\_symbols\_count\_matrix = zeros(|alphabet|, 1)$
15     for $i \in \{0, \ldots, |alphabet| - 1\}$ do
16         $i_s = 0.0$
17         for $j \in \{0, \ldots, |alphabet| - 1\}$ do
18             $i\_s = i\_s + i\_symbols\_j\_count\_matrix[i][j]$
               $s = s + i\_symbols\_j\_count\_matrix[i][j]$
19         $i\_symbols\_count\_matrix[i] = i\_s$
20     // Build s.j
21     $symbols\_j\_count\_matrix = zeros(|alphabet|, 1)$
22     for $j \in \{0, \ldots, |alphabet| - 1\}$ do
23         $s_j = 0.0$
24         for $i \in \{0, \ldots, |alphabet| - 1\}$ do
25             $s\_j = s\_j + i\_symbols\_j\_count\_matrix[i][j]$
26         $symbols\_j\_count\_matrix[j] = s\_j$
```

The *RissanenESI*() function is continued on the following page.

Algorithm 4: (Continued) The *RissanenESI*() function on one node.

```
1  begin
2      // Initialize the ESI value
3      $ESI\_sum = 0$
4      $leaf\_distribution = \{\}$
5      for $i \in \{0, \ldots, |alphabet| - 1\}$ do
6          for $j \in \{0, \ldots, |alphabet| - 1\}$ do
7              $i\_s\_j = i\_symbols\_j\_count\_matrix[i][j]$
8              $i\_s = i\_symbols\_count\_matrix[i]$
9              $s\_j = symbols\_j\_count\_matrix[j]$
10             $ESI\_sum = ESI\_sum + (i\_s\_j) * log_2((i\_s\_j/i\_s)/(s\_j/s))$
11         $s\_j = symbols\_j\_count\_matrix[i]$
12         // Save the array as a string in order to be saved to a file later on
           $leaf\_distribution[alphabet[i]] =$
           $process\_numpy\_array\_value\_to\_string(s_j/s)$
13     return $[ESI\_sum, leaf\_distribution]$
```

These set of algorithms are always started with Stage 1 in order to build up the contexts. At the end of each stage the contexts are collected from all the nodes and stored into one file, which will be used as an input for the following stage. This writing of the files by each node is equivalent to the Map phase of the MapReduce algorithm, while the collection is the Reduce phase of it. In order to build the final SCOT file to a specific stage, all the contexts to the desired stage are combined and the horizon is consolidated with all the non-contexts of only the specific stage.

The objects for parallelism in the SCOT construction are substrings, which are tested for being contexts via the *RissanenESI* function. Such potential contexts would require the prefix and suffix counts to be populated. To parallelize the processing of contexts, a hash function is implemented to construct a numerical value out of the characters forming the context string, which subsequently is modulo with the number of nodes (*total_nodes* variable). This way each compute node will process a portion of the potential contexts, which can be many in latter stages of a SCOT. The hashing algorithm for strings (Algorithm 5) — which is described on the following page — is a common one (Kernighan and Ritchie).

Algorithm 5: The *hash_of_sequence*() function on one node.

Inputs:
sequence_of_symbols: The sequence of symbols to hash.

Outputs:
Returns the hash of the *sequence_of_symbols*.

```
1 begin
2     // A large prime number
3     $number\_of\_bins = 99971$
4     $hash\_key = 0$
5     $odd\_integer\_multiplier = 31$
6     for $i \in \{0, \ldots, |sequence\_of\_symbols|$ do
7         $hash\_key = ((hash\_key * odd\_integer\_multiplier) + get\_ASCII\_value(sequence\_of\_symbols[i]))\ mod\ number\_of\_bins$
8     return $hash\_key$
```

The hash function is applied by the following algorithm, in order to determine if a sequence should be processed on a specific node out of many allocated ones:

Algorithm 6: The *should_context_be_processed_on_this_node*() predicate function to determine if a context should be processed on the current node.

Inputs:
sequence_of_symbols: The sequence of symbols to determine it should be processed.
node_id: Node ID out all nodes.
total_nodes: Total number of compute nodes.

Outputs:
Returns True if *sequence_of_symbols*, otherwise False.

```
1 begin
2     $hash\_key = hash\_of\_sequence(context\_to\_check)$
3     return $((hash\_key \ \% \ total\_nodes) == node\_id)$
```

The medium-sized prime number 99971 is used to distribute the storage of data for load-balancing the retrieval of sequence counts. Each sequence thus would be stored in a list (chained) in the bucket denoted by the modulo — because of possible collisions — which would look as follows:

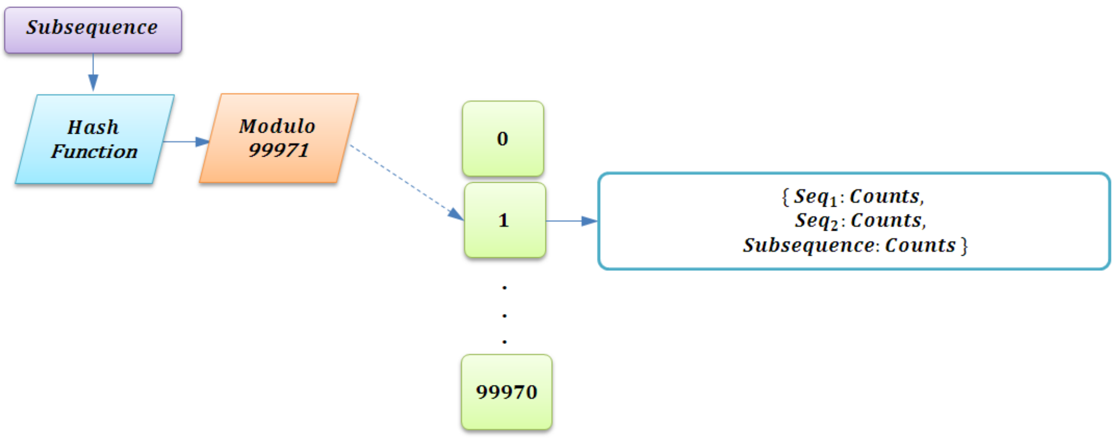


Figure 4: The process by which a subsequence is stored into the hash-based bucket data-structure for optimal retrieval.

The *populate_NonContextSequenceCounts_and_FilterOutEstablishedContexts_in_Corpus* in the *run_stage* performs a check — via a predicate helper function — on the potential context to determine if it should be processed on the compute node as follows ($(hash\_key \ \% \ total\_nodes) == node\_id$). If the potential context is run on a particular node then a helper function performs an initial check by comparing the potential context subsequence against the keys of known previously determined contexts. If it is a context then the prefix and suffix counts encountered across the corpus for such a subsequence is not performed. Otherwise, that count is then incremented accordingly.

8.3. *Homogeneity Testing For Authorship Attribution*

We now describe the Homogeneity Test using a SCOT model and two corpora for authorship attribution outlined at the end of section 7. The idea is to splice each corpus into k slices and using the SCOT walk along each segment character-wise. At each position, that will be considered the future character and by walking backward on subsequences of characters previous to it, we search for the corresponding SCOT context. If one is found then we retrieve the probability of the future character in the leaf-distribution, otherwise return a small probability of 10^{-10}. For each segment, the log of these probabilities will be evaluated and summing the logs of these probabilities we to obtain the log-likelihood of that segment. Since each segment does not have a significant past of characters at the beginning, the last 50 characters of the corpus will be attached to it for evaluation. As the slice progresses more of the sequence will be related to its corpus.

Let us apply the theory of the section 7 to the homogeneity testing of multivariate distributions of the large strongly stationary ergodic training string T and a query string Q. We use the nonparametric test of (Malyutov et al., 2013). The first stage is estimation of the SCOT model of the string T following the algorithm in (Mächler and Bühlmann, 2004). We refer to this publications and our section 7 for details.

We summarize our test statistics by the following equation:

$$Homogeneity\ Test = \left(\underset{k}{\text{maximize}} \frac{(\sum L^T - \sum L^Q)/\sqrt{k}}{\sqrt{VAR(L^T) + VAR(L^Q)}} \right)$$

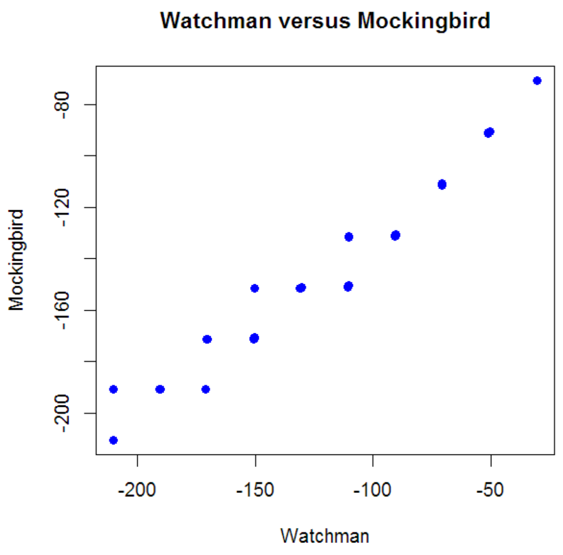
The key function that performs the likelihood calculation is called *get_log_likelihoods*() and described in the following pages, which was written using the Python programming language.

Algorithm 7: The *get_log_likelihoods*() function on one node.

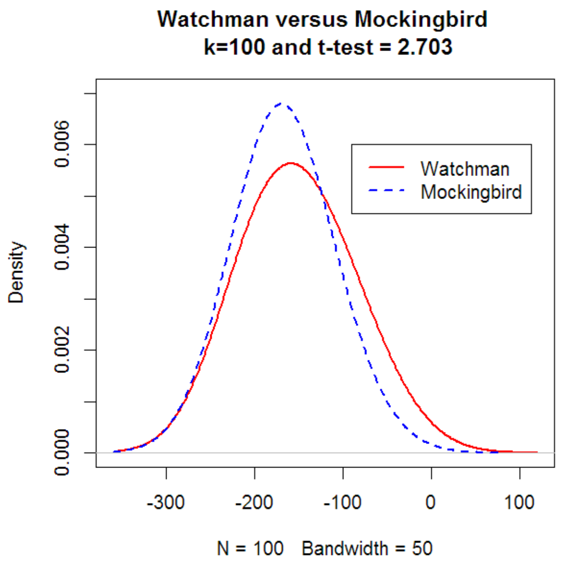
```
Inputs:
$Text\_corpus\_file$: The corpus a text.
$Text\_SCOT$: The SCOT trained a text.
$k\_number\_of\_segments$: k segments to splice the corpora in order to
calculate the likelihood for each.
$node\_id$: Node ID out all nodes.
$total\_nodes$: Total number of compute nodes.

Outputs:
Returns the log-likelihoods of segments run a node.

1  begin
2      $log\_likelihoods = []$
3      // The segment index starts at 1 and proceeds to k
4      for $i \in \{1, \ldots, k\_number\_of\_segments\}$ do
5          $sequence\_segment =$
           $get\_segment(Text\_corpus, i, k\_number\_of\_segments)$
6          // The hash of the string with the modulo of the $total\_nodes$ is
           performed here
7          if $should\_sequence\_be\_processed\_on\_this\_node(sequence\_segment,$
8                                       $node\_id, total\_nodes)$ then
9              $log\_likelihood\_within\_segment = 0.0$
10             for $i \in \{0, \ldots, |sequence\_segment| - 1\}$ do
11                 $log\_likelihood\_within\_segment =$
                   $log\_likelihood\_within\_segment +$
                   $log(get\_future\_symbol\_likelihood(Text\_corpus,$
12                                 $sequence\_segment, i, Text\_SCOT))$
13             $log\_likelihoods.append(log\_likelihood\_within\_segment)$
14     return $log\_likelihoods$
```

The corpora are limited to ensure equal length and equal segment sizes. We ran a test of the first 1000 characters with $k = 100$ after post-processing of Chapter 1 of *Go Set a Watchman* against Chapter 1 of *To Kill A Mockingbird*, both by Harper Lee to test style homogeneity of these novels. The SCOT was trained against *Go Set a Watchman* to stage

15 with an $\epsilon = 3$. We chose these settings in order to maximize finding contexts in the SCOT in order to best determine the chosen contexts over this large period of time. The t-test result was 2.70299491543 which shows significance of style difference. We generated the q-q plot and distribution to illustrate the results:



The q-q plot of Watchman versus Mockingbird.

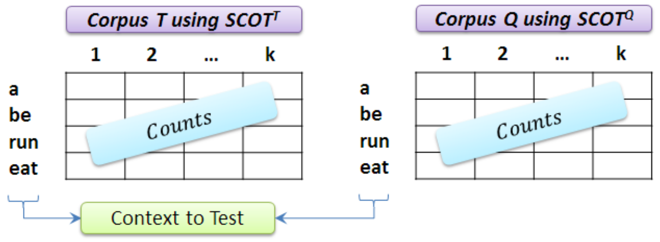


The distributions of Watchman versus Mockingbird.

Our experience shows that attempts to deliberately change style usually do not lead to significant changes of microstyle. Our discrimination method apparently uses unconscious phonetic habits. This should be confirmed by further studies.

8.4. *Follow-Up Analysis of the the Most Contributing Patterns*

In case there is a significant difference between corpora in the Homogeneity test, then using the number of slices k derived above we then determine the most contributing patterns by running $Corpus^T$ against $SCOT^T$, and $Corpus^Q$ against $SCOT^Q$. These generate counts for each context in each k-segment populating the following matrices:



The matrices used to populate the contexts in each segment of each corpus (given the corresponding SCOT process) are stored into the hash-based bucket data-structure for optimal retrieval.

These counts are generated via the following four statistics per context i: $E[Context_i^T]$, $E[Context_i^Q]$, $VAR[Context_i^T]$, and $VAR[Context_i^Q]$. These will be then be used to get the maximum contributing patterns (CP) in T and Q as follows:

$$CP^T = \left(\underset{i}{\text{maximize}} \frac{(E[Context_i^T] - E[Context_i^Q])/\sqrt{k}}{\sqrt{VAR(Context_i^T) + VAR(Context_i^T)}} \right)$$

$$CP^Q = \left(\underset{i}{\text{minimize}} \frac{(E[Context_i^T] - E[Context_i^Q])/\sqrt{k}}{\sqrt{VAR(Context_i^T) + VAR(Context_i^T)}} \right)$$

Such statistics are sorted by the respective CP^T and CP^Q to determine the most contributing patterns in each corpora. As is previously described, the parallelism applied here on each node is by performing the hash of the context with the modulo of the *total_nodes*, in order that it should be processed by the node.

Our results are available by request.

9. Discussion, open problems and acknowledgments

Our presentation on modeling and asymptotic inference of strongly mixing stationary sequences differs drastically from the material presented in traditional courses on stationary processes and connects this discipline with the classical MC-theory.

Our AN derivation for ATF and of LAN property of SCOT models is new and seems transparent. Parallelism in training SCOT enables possibility to run it with various thresholds for the ESI-criterion and choosing most relevant one. This was only asymptotically studied before.

A formalization of convergence of strongly mixing stationary sequences to m-MC remains the main task to clarify.

Another challenge is the relation of the memory-spectrum and the entropy-based approaches for characterizing the sparsity of approximating m-MC.

The main remaining hard asymptotic problem is to prove accurate asymptotic results for the case of the number of SCOT contexts rising simultaneously with the sample size.

We are grateful to Professor Cooperman and the National Science Foundation, NSF Award OCI 12-29059, "MRI Consortium: Acquisition of a heterogeneous, shared, computing instrument to enable science and computing research by the Mass. Green High performance Computing Consortium." We are also grateful to V. N. Tutubalin for the useful discussion.

References

SLURM: Simple Linux Utility for Resource Management. https://computing.llnl.gov/linux/slurm/.

P. Billingsley. *Statistical inference for Markov chains.* University of Chicago Press, 1961.

E. Bolthausen. The Berry-Esseen theorem for functionals of discrete Markov chains. *Z. Wahrsch. Verw. Gebiete*, 54:59–73, 1980.

E. Bolthausen. The Berry-Esseen theorem for strongly mixing Harris recurrent Markov chains. *Z. Wahrsch. Verw. Gebiete.*, 60:283–289, 1982.

A. A. Borovkov. *Ergodicity and stability of stochastic processes.* Wiley, 1998.

R. C. Bradley. A caution on mixing conditions for random fields. *Statist. Probab. Letters*, 8:498–491, 1989.

R. C. Bradley. Basic properties of strong mixing conditions. A survey and some open questions. *Probability Surveys*, 2:107–144, 2005.

P. Bühlmann and A. Wyner. Variable length Markov chains. *The Annals of Statistics*, 27 (2):480–513.

D. M. Chibisov. Lectures on the asymptotic theory of rank tests. *Lecture Notes NOTs 14. M.: Matematicheskiy Institut im. V. A. Steklova, RAN*, 2009. In Russian.

T. M. Cover and J. A. Thomas. *Elements of information theory.* Hoboken: Wiley, second edition, 2006.

J. Dean and S. Ghemawat. MapReduce: simplified data processing on large clusters. *Proceedings of the 6th conference on Symposium on Operating Systems Design & Implementation*, 6:1–13, 2004. (OSDI'04), USENIX Association.

A. D. Deyev. Asymptotic expansion of classification statistics in normal case, 1970.

W. Feller. *An introduction to Probability theory and its applications*, volume 1. Wiley, N. Y., third edition, 1967.

W. Feller. *An introduction to Probability theory and its applications*, volume 2. Wiley, N. Y., second edition, 1970.

A. Galves and E. Loecherbach. Stochastic chains with memory of variable length. *TICSP series*, (38):117–134, 2008. In: Festschrift in Honor of Jorma Rissanen on the Occasion of his 75th Birthday, Tampere Tech. Uni.

C. M. Grinstead and J. L. Snell. *Introduction to Probability.* AMS, 2006.

J. L. Jensen. Asymptotic expansions for strongly mixing harris recurrent markov chains. *Scandinavian Journal of Statistics*, 16(1):47–63, 1989.

B. W. Kernighan and D. M. Ritchie. *The C Programming Language.* Prentice Hall, 2nd edition.

M. Mächler and P. Bühlmann. *Variable Length Markov Chains: methodology, computing, and software*, volume 13. 2004.

V. K. Malinovsky. On limit theorems for Harris Markov chains, I. *Theory Probab. Appl.*, pages 269–285, 1987. (English translation).

M. Malyutov and R. Protassov. LAN and LAM: Convergence of Iterative Estimates and Optimal Design in Gaussian One-Way Mixed Model. *Journal of Statistical Planning and Inference*, 100(2):249–279, 2002.

M. B. Malyutov and T. Zhang. Limit theorems for additive functions of SCOT trajectories. *Information Processes*, 15(1):89–96, 2015.

M. B. Malyutov, T. Zhang, X. Li, and Y. Li. Time series homogeneity tests via VLMC training. *Information Processes*, 13(4), 2013.

S. P. Meyn and R. L. Tweedy. *Markov chains and stochastic stability.* Springer, 1993.

J. Rissanen. A universal data compression system. *IEEE Trans. Inform. Theory*, 29(5): 656–664, 1983.

G. Roussas. *Contiguity of probability measures: some applications in statistics.* Cambridge University Press, 1972.

B. Ryabko, J. Astola, and M. Malyutov. *Compression-Based Methods of Statistical Analysis and Prediction of Time Series.* Springer International Publishing AG Switzerland, 2016.

V. N. Tutubalin. *Probability and random processes theory: Mathematical foundations and applications.* Moscow State University Press, 1992. (In Russian).

A. Veretennikov. *Parametric and nonparametric estimation for Markov Chains.* Moscow State University Press, 2000. (In Russian).

V.A. Volkonskii and Yu. A. Rozanov. Some limit theorems for random functions I. *Theor. Probab. Appl.*, 4:178–197, 1959.

T. Zhang. Perfect Memory Context Trees in time series modeling. 2016. *arXiv:1610.08910v1 [cs.LO].*