Computational Statistics and Data Analysis 70 (2014) 281–294

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

A graph theoretic approach to simulation and classification

Michael A. Kouritzin, Fraser Newton *, Biao Wu

Department of Mathematical and Statistical Sciences, University of Alberta, Edmonton, T6G 2G1, Canada

ARTICLE INFO

Article history:
Received 17 November 2012
Received in revised form 24 September 2013
Accepted 24 September 2013
Available online 11 October 2013

Keywords:
Optical character recognition
Random field
Graph theory
Spatial correlation
Simulation

ABSTRACT

A new class of discrete random fields designed for quick simulation and covariance inference under inhomogeneous conditions is introduced and studied. Simulation of these correlated fields can be done in a single pass instead of relying on multi-pass convergent methods like the Gibbs Sampler or other Markov chain Monte Carlo algorithms. The fields are constructed directly from an undirected graph with specified marginal probability mass functions and covariances between nearby vertices in a manner that makes simulation quite feasible yet maintains the desired properties. Special cases of these correlated fields have been deployed successfully in data authentication, object detection and CAPTCHA[1] generation. Further applications in maximum likelihood estimation and classification such as optical character recognition are now given within.

1. Introduction

Random fields are widely used in sciences and technologies to model spatially distributed random phenomena or objects. Within science, random fields are used in geophysics, astrophysics, statistical mechanics, underwater acoustics, structural biology and agriculture. Applications of random fields in technologies include TV signal processing, image processing in photography such as medical images (human brain imaging, functional magnetic resonance imaging, mammography), computer vision, web data extraction, clustering gene expression time series, natural language processing, etc. Readers are referred to Ashburner et al. (2003), Chellappa and Jain (1993), Li et al. (2008, 1995), Li (1995), Winkler (2003), Worsley (1995), Zhang et al. (2001), and Zhu et al. (2008) for those applications. Technologically, researchers of random fields have dealt either with the modeling of images (for synthesis, recognition or compression purposes) or with the resolution of various spatial inverse problems (image restoration and reconstruction, deblurring, classification, segmentation, data fusion, optical flow estimation, optical character recognition, stereo matching, finger print classification, pattern recognition, face recognition, intelligent video surveillance, sparse signal recovery, natural language processing like Chinese chunk and so on, see Blue et al. (1993), Chellappa et al. (1995), Li (1995), Sun et al. (2008), and Winkler (2003)).

Scientists and technicians are interested in the inverse problems such as image restoration, boundary detection, tomographic reconstruction, shape detection from shading, and motion analysis. Many precisely formulated mathematical models were constructed to model certain types of random fields, and various methods and estimators have been developed to make the proposed models work in application. There are diverse needs calling for simulating random fields. For example, simulation is employed to calculate minimum mean square (MMS) and maximum posterior marginal (MPM) estimators, see Winkler (2003). Simulation can also be a smoothing technique. In Chapter 2 of Winkler (2003), various smoothing techniques were proposed to clean "dirty" pictures. Most of these methods involve simulation. The difficult problem is how to simulate

* Corresponding author. Tel.: +1 7804459058.
E-mail address: fenewton@ualberta.ca (F. Newton).

[1] An acronym for "Completely Automated Public Turing test to tell Computers and Humans Apart" that is widely used to protect online resources from abuse by automated agents.

0167-9473/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.csda.2013.09.026

random fields effectively. A typical simulation would involve many correlated random variables, and could easily exceed the capacity of modern computers if one tried to simulate the whole random field from the probability distribution directly.

Researchers frequently resort to imposing discrete Markov assumptions on their random fields to be simulated out of practical need. In this regard, the Gibbs sampler was proposed to ease this simulation difficulty. Briefly speaking, a Gibbs sampler starts with a given *initial configuration* (i.e. potential realization of the random field) or a configuration chosen at random from some initial distribution, and then updates its configuration vertex by vertex based on the local characteristics of the random field. Once all vertices of a configuration are sequentially updated, a *sweep* or a *pass* is finished. A Gibbs sampler usually takes hundreds of sweeps to produce a configuration closely consistent with a given distribution and there are still computational and convergence issues to deal with. Indeed, the number of possible random configurations within a general discrete random field can be enormous and simulation is further complicated when the vertices are correlated with one another. These factors can make Gibbs sampling and other Markov chain Monte Carlo simulation impractical.

In this paper, we propose a graph theoretic construction for simulation and introduce a new class of discrete *correlated random fields* which incorporate given probability mass functions (pmfs) corresponding to vertices in a graph and pairwise covariances corresponding to edges in a graph. These fields are designed with efficient simulation in mind. Proposition 1 on which our fields are based establishes a method to imbed desired covariances and marginal probabilities into a random field while maintaining simulation ease. Indeed, Proposition 1 is a simple means to construct *some* conditional probabilities consistent with given marginal probabilities and covariances in such a way that sampling the missing portion of a random field sequentially is very feasible. More precisely, when simulating a new vertex, we compute this conditional probability of its state conditioned on the known portion and the previously-simulated vertices. We can construct a random field in one pass based on this algorithm. This method is especially suited to problems where pairwise covariance captures the meaningful relationships between variables. (We explore the role of covariances further in Section 4.) For demonstration purposes, we discuss prior applications of our random fields to data authentication, object detection and CAPTCHA generation, as well as develop new example applications in maximum likelihood estimation and classification. In particular, our experimental results suggest our algorithm may help improve optical character recognition.

The remainder of this note is laid out as follows. Section 2 contains our notation and the statement of our main results, Proposition 1; Section 3 provides several mathematical examples illustrating the method; and in Section 4, we develop new applications in maximum likelihood estimation and optical character recognition.

2. Notation and background

Our goal is to simulate a random field so that desired properties (in our case, marginal probabilities and pair-wise covariances) are maintained. Specifically, we will give a method for computing conditional probabilities so that these properties are maintained. We begin by describing how the problem is constructed in Section 2.1; then, we will describe exactly how to compute the probabilities in Section 2.2; finally, illustrative examples are given in Section 3.

2.1. Problem setup

Suppose we are given a desired probability mass function (pmf) for each random variable and a set of desired pairwise covariances for some set of pairs of the random variables. Our goal is to simulate the random variables so that the desired properties are met.

2.1.1. Definitions

We will be working with undirected and directed graphs in the following. We begin by providing the required definitions and notation.

An *undirected graph* $G = (V, E)$ is a set of vertices V and edges E between some of the vertices, where $(u, v) \in E$ if there is an edge between vertices u and v; in this case, u and v are called *neighbors*. A *directed graph* $D = (V, A)$ is a set of vertices V and arcs A from some of the vertices to others, where $(u, v) \in A$ if there is an arc from vertex u to vertex v; in this case, u is called a *parent* of v and v is called a *child* of u. More generally, we would say that v is an ancestor of u if there are a collection of arcs starting at v and going to u such that the first arc starts at v, the last arc ends at u and every arc in between starts where the previous one ends. The real difference between directed and undirected graphs is the former has a direction to its edges.

An undirected graph is called *connected* if there is a path of edges between every pair of vertices. The *open neighborhood* $N_G(v)$ of vertex $v \in V$ is the set of vertices $u \neq v$ such that there is an edge between u and v, i.e., $(u, v) \in E$. We denote the open neighborhood of v by $N_G(v)$ and the closed neighborhood $N_G(v) \cup \{v\}$ by $N_G[v]$. For a set of vertices B, we define the open neighborhood of B as $N_G(B) = \cup_{v \in B} N_G(v) \setminus B$ and closed neighborhood $N_G[B] = N_G(B) \cup B$. For convenience, we set $N_G(\emptyset) = V$.

A directed graph is called *acyclic* if there is no vertex v that is an ancestor of itself.