MAXENTMC – A MAXIMUM ENTROPY ALGORITHM WITH MOMENT CONSTRAINTS

RAFAIL V. ABRAMOV

Contents

1. Introduction

The MaxEntMC is a maximum entropy algorithm with moment constraints, which can be adapted to computation of the maximum entropy problem with the user-defined set of moment constraints, user-defined quadrature (and, therefore, domain) on the N-dimensional Euclidean space (where N is realistically no greater than 3 or 4 due to computational limitations on the numerical quadrature), via user-defined iterations of finding the critical point of the corresponding objective function (such as the Newton or quasi-Newton methods). In a certain sense, the MaxEntMC is a set of tools to produce an individually tailored maximum entropy algorithm for a given problem, although some simple examples of maximum entropy quadrature and iteration implementations are included with the code. The MaxEntMC provides the user with the following core tools for the maximum entropy computation:

(1) A set of data structures and routines to initialize, save and load moment constraints and corresponding Lagrange multipliers of arbitrary dimension and power;
(2) A set of quadrature helper routines to implement a parallel (currently shared-memory multithreaded) numerical quadrature for density moments over a set of user-specified abscissas and weights;
(3) A set of routines to transform the computed density moments into the gradient vector and hessian matrix of the Lagrangian objective function.

In addition, the MaxEntMC provides the user with the following example routines for better illustration:

(1) A simple uniform quadrature routine over a user-specified rectangular domain;
(2) A Newton method routine with a primitive inexact line search for computation of the critical point of the objective Lagrangian function.

It is, however, expected that the user will tailor the algorithm precisely to their particular maximum entropy problem by implementing their own quadrature and iteration methods.

The manual is organized as follows. Section 2 presents a mathematical formulation of the maximum entropy problem with moment constraints. Section 3 documents the routines necessary for operations with moment constraints and Lagrange multipliers. Section 4 documents the quadrature helper routines which are needed to set up the user-specified quadrature. Section 5 documents the routines which are used to extract the gradient and hessian of the Lagrangian function from the computed moments in a generic vector and matrix form.

2. Mathematical formulation of the maximum entropy problem with moment constraints

The content of this section is based on my paper titled *"The multidimensional maximum entropy moment problem: A review on numerical methods"*, published in Commun. Math. Sci. Some parts of the section content below may be copied, with minor changes, from that work. For more details, please refer to that work and references therein.

2.1. **Notations.** Let N be a positive integer number. Let $\mathbb{R}^N$ be an N-dimensional space of real numbers, and let $\mathbb{Z}^N$ be an N-dimensional space of nonnegative integer numbers. Let $\boldsymbol{x}$ denote an element of $\mathbb{R}^N$, and let $\boldsymbol{i}$ denote an element of $\mathbb{Z}^N$. We define a *monomial* $\boldsymbol{x}^{\boldsymbol{i}}$ as

$$\boldsymbol{x}^{\boldsymbol{i}} = x_1^{i_1} x_2^{i_2} \ldots x_N^{i_N} = \prod_{k=1}^{N} x_k^{i_k}. \tag{2.1}$$

We denote the *power* $|\boldsymbol{i}|$ of the monomial in (2.1) as

$$|\boldsymbol{i}| = i_1 + i_2 + \ldots + i_N = \sum_{k=1}^{N} i_k. \tag{2.2}$$

Let U be a domain in $\mathbb{R}^N$. Let $\rho : U \to \mathbb{R}$ denote any nonnegative continuous function on U, with the condition that

$$\int_U \rho(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} < \infty. \tag{2.3}$$

In this case, ρ is called the *density*. Given an element $\boldsymbol{i} \in \mathbb{Z}^N$, we denote the corresponding *moment* $m_{\boldsymbol{i}}[\rho]$ as

$$m_{\boldsymbol{i}}[\rho] = \int_U \boldsymbol{x}^{\boldsymbol{i}} \rho(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}. \tag{2.4}$$

Let P be a finite subset of $\mathbb{Z}^N$. Let $\rho_{\boldsymbol{c}} : U \to \mathbb{R}$ denote any ρ for which all $m_{\boldsymbol{i}}$, such that $\boldsymbol{i} \in P$, are finite, and are given by a set of real numbers $c_{\boldsymbol{i}} \in \mathbb{R}$, for all $\boldsymbol{i} \in P$:

$$m_{\boldsymbol{i}}[\rho_{\boldsymbol{c}}] = c_{\boldsymbol{i}}, \qquad \forall \boldsymbol{i} \in P. \tag{2.5}$$

It is then said that $\rho_{\boldsymbol{c}}$ possesses a set of moments (or *moment constraints*, in the context of what is presented below) $\boldsymbol{c}$. There are two things which need to be emphasized:

(1) For an arbitrarily specified set of real numbers $\boldsymbol{c}$, the corresponding density $\rho_{\boldsymbol{c}}$ does not have to exist (for example, suppose that $\boldsymbol{i}$ includes only even integers, yet $c_{\boldsymbol{i}} < 0$);
(2) When $\rho_{\boldsymbol{c}}$ nonetheless exists, it does not have to be unique.

What is presented below deals with the second situation.

2.2. **Maximum entropy under moment constraints.** We define the *Shannon entropy* $S[\rho]$ as

$$S[\rho] = -\int_U \rho(\boldsymbol{x}) \ln \rho(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}. \tag{2.6}$$

Shannon entropy is recognized as a measure of uncertainty in probability densities. In the context of the maximum entropy problem, the goal is to find $\rho_{\boldsymbol{c}}^*$ among all $\rho_{\boldsymbol{c}}$, such that is maximizes S:

$$\rho_{\boldsymbol{c}}^* = \arg\max_{\rho_c} S[\rho_{\boldsymbol{c}}]. \tag{2.7}$$

It can be shown via variational analysis that $\rho_{\boldsymbol{c}}^*$ belongs to the family of functions $f_{\boldsymbol{\lambda}} : U \to \mathbb{R}$, where $\boldsymbol{\lambda} = \{\lambda_{\boldsymbol{i}}\}$ are real numbers (called the *Lagrange multipliers*), $\boldsymbol{i} \in P$. $f_{\boldsymbol{\lambda}}(\boldsymbol{x})$ is explicitly given by

$$f_{\boldsymbol{\lambda}}(\boldsymbol{x}) = \exp\left(\sum_{\boldsymbol{i} \in P} \lambda_{\boldsymbol{i}} \boldsymbol{x}^{\boldsymbol{i}}\right). \tag{2.8}$$

Depending on the sets U, P and on the values $\boldsymbol{\lambda}$, $f_{\boldsymbol{\lambda}}(\boldsymbol{x})$ may or may not be a density; indeed, observe that while any $f_{\boldsymbol{\lambda}}$ is certainly nonnegative, its integral over U does not have to be finite. More precisely, any $f_{\boldsymbol{\lambda}}$ is a density over a finite domain U, but not all $f_{\boldsymbol{\lambda}}$ are densities when U is not a finite domain.

Our goal now is to find the set $\boldsymbol{\lambda}^*$ for which

$$f_{\boldsymbol{\lambda}^*}(\boldsymbol{x}) = \rho_{\boldsymbol{c}}^*(\boldsymbol{x}), \qquad \forall \boldsymbol{x} \in U. \tag{2.9}$$

This is accomplished by computing the minimum of the *Lagrangian function* (or shortly, the Lagrangian)

$$L(\boldsymbol{\lambda}) = \int_U f_{\boldsymbol{\lambda}}(\boldsymbol{x}) \, \mathrm{d}x - \sum_{\boldsymbol{i} \in P} c_{\boldsymbol{i}} \lambda_{\boldsymbol{i}}. \tag{2.10}$$

The gradient (the vector of the first derivatives) and hessian (the matrix of the second derivatives) of the Lagrangian are given, respectively, via

$$\frac{\partial L}{\partial \lambda_{\boldsymbol{i}}} = m_{\boldsymbol{i}}[f_{\boldsymbol{\lambda}}] - c_{\boldsymbol{i}}, \tag{2.11a}$$

$$\frac{\partial^2 L}{\partial \lambda_{\boldsymbol{i}} \partial \lambda_{\boldsymbol{j}}} = m_{\boldsymbol{i}+\boldsymbol{j}}[f_{\boldsymbol{\lambda}}]. \tag{2.11b}$$

From the above expressions, it is easy to see that the gradient of L is zero when the moment constraints are met (that is, a critial point of L is found), and that the hessian of L is positive definite, which means that the critical point is indeed a unique minimum. Note that, in general, the critial point does not have to exist.

The optimal set of Lagrange multipliers $\boldsymbol{\lambda}$ can be found iteratively, using a plethora of standard methods such as the Newton method, or a variable metric quasi-Newton method (such as the BFGS algorithm). The Newton method requires the computation of the hessian, while the quasi-Newton methods use a "finite difference" secant approximation for the hessian from the set of gradients computed at different points along the optimization path. Either iteration method can be implemented with the MaxEntMC.

3. How to specify constraints and Lagrange multipliers

To be written.

4. How to compute moments

To be written.

5. How to compute the gradient and hessian of the objective function

To be written.