Chapter 1
Constrained density estimation

Peter Laurence, Ricardo J. Pignol, and Esteban G. Tabak

Abstract
A methodology is proposed for non-parametric density estimation, constrained by the known expected values of one or more functions. In particular, prescribing the first moment –the mean of the distribution– is a requirement for the density estimation associated to martingales. The problem is addressed through the introduction of a family of maps that transform the unknown density into an isotropic Gaussian, while adjusting the prescribed moments of the estimated density.

1.1 Introduction

Density estimation, a general problem arising in many applications, consists of inferring the probability distribution $\rho(x)$ underlying a set of independent observations $x_j \in \mathbb{R}^n$, $j = 1,\ldots,N$. Extra information on the distribution $\rho(x)$ is often available, in addition to the N observations x_j, in the form of the expected values $\bar{f}_i$ of q functions $f_i(x)$, $i = 1,\ldots,q$. The availability of these expected values may be due to a variety of reasons:

- They may be independently observed. In the natural sciences, for instance, a number of devices and experimental procedures are designed to measure the mean value of a quantity over a large population, an extended area or a fluctuating field. In the financial markets, the pricing of options provide information on the expected values of future prices of the underlying assets under the risk-neutral measure.
- Due to storage or technological limitations, or to the specific interests of the parties involved, historical records or records from individual laboratories or polling agents may have only kept the mean values of certain functions of interest.
- They may arise from theoretical considerations. Economic theory, for instance, requires the conditional probability of the risk-neutral measure underlying a time series x_t of prices to be a martingale, which imposes a condition on its mean:

$$\mathrm{E}\,(X_t|X_{t-1},X_{t-2},\ldots) = X_{t-1}.$$

  Another example is the requirement that the support of the distribution sought should not contain certain scenarios, a constraint that can be phrased in terms of expectations:

Universitá di Roma I, Universidad Nacional del Sur and Courant Institute

$$\mathrm{E}\left(I_U(X)\right) = 0, \tag{1.1}$$

where I_U is the indicator function of the set U of excluded events.

Then the following constrained density estimation problem arises: given a set of observations $x_j \in \mathbb{R}^n$, $j = 1,\dots,N$, estimate its probability density $\rho(x)$, subject to the constraints that the expected value of q functions f_i are prescribed:

$$\mathrm{E}(f_i(X)) = \bar{f}_i, \quad i = 1,\dots,q. \tag{1.2}$$

Here f_i are real-valued functions defined on a domain in $\mathbb{R}^n$. In the terminology of financial engineering, this may be referred to as *calibration* of the pricing distributions for consistency with available market data. For instance in energy markets, a common hedging instrument is a spread option, with payoff $S_2 - S_1 - K$, where $S_{1,2}$ are the prices of two assets, such as crude and refined oil for the crack spread option, and K is the option's strike price. The price of such an option depends on the joint distribution of the assets under the risk neutral measure. Typically the market provides a lot of information on the marginal distributions of the two assets, via liquidly traded options with many strikes and maturities on each asset S_1 and S_2. On the other hand a much more limited number of strikes trade on the spread $S_2 - S_1$, so that the market provides only limited information of the *joint distribution* of the two assets. We will return to this example in the numerical section.

More generally, the estimated density may also need to satisfy inequality constraints, of the form

$$\mathrm{E}(f_i(X)) \le \bar{g}_i, \quad i = 1,\dots,p. \tag{1.3}$$

This is the case, for instance, when at least a certain fraction of the probability is required to lie in the tail of the distribution, and when the density $\rho(x)$ itself is known to be bounded above by a function $h(x)$, a condition that admits the weak formulation

$$E(\phi(X)) \le \int \phi(x)\, h(x)\, dx$$

for all smooth positive functions $\phi(x)$, thus involving infinitely many inequality constraints of the form (1.3).

The methodology developed in this paper builds, for a given set of observations of a random variable x in $\mathbb{R}^n$, a sequence of parametric maps that take the unknown probability distribution function $\rho(x)$ to an isotropic Gaussian $\mu(y)$. A procedure along these lines was developed in [1, 2] for unconstrained density estimation, using the composition of simple maps that increase the log-likelihood of the data at each step. By keeping track point-wise of the compounded map and its Jacobian, one can reconstruct the unknown probability density function underlying the observations, evaluating it on the observed values themselves and on any other pre-determined set of values of x, that the algorithm carries passively through the maps.

The estimated density takes the form

$$\rho(x) = J_y(x)\ \mu(y(x)), \tag{1.4}$$

where $J_y(x)$ is the Jacobian determinant of the map $y(x)$. The constraints on the density $\rho(x)$ in (1.2), (1.3) become, in view of (1.4), constraints on the allowed maps $y(x)$. The transformation $y(x)$ is built gradually, through the composition of near identity maps. Thus we build a sequence $y_k(x)$, where

$$y_{k+1}(x) = z_k\left(y_k(x)\right), \tag{1.5}$$

with

$$z_k(y) = y + \varphi_k(y), \qquad \|\varphi_k(y)\| \ll 1. \tag{1.6}$$

Correspondingly, there is a sequence of Jacobian determinants $J_k(x)$, with

$$J_{k+1}(x) = j_k(y_k(x))J_k(x), \tag{1.7}$$

where $j_k(y)$ is the Jacobian of $z_k(y)$.

The algorithm alternates between two kinds of steps: in the first , the functions (maps) $\varphi_k(y)$ are chosen so as to move the distribution $\rho(x)$ towards satisfying the constraints in (1.2, 1.3). In the second, the maps are chosen so as to improve the log-likelihood of $\rho(x)$ on the data points x_j, while not deteriorating the current level of accommodation of the constraints.

1.2 The two objective functions

The procedure is based on two objective functions: the log-likelihood of the data

$$L = \sum_j \log\left(\rho\left(x_j\right)\right) \tag{1.8}$$

that one attempts to maximize, and a cost associated with the non-satisfaction of the constraints,

$$C = \frac{1}{2}\sum_i w_i\, C_i^2, \quad w_i > 0 \tag{1.9}$$

that one seeks to minimize. Here C_i is a Montecarlo estimate to be detailed below of the difference between the current estimation for the expected value of $f_i(x)$,

$$E(f_i(x)) = \int f_i(x)\ \rho(x)\ dx, \tag{1.10}$$

and its prescribed value $\bar{f}_i$. The weights w_i have a dual purpose: to normalize the range or variability around each $\bar{f}_i$, and to qualify the level of significance attached to each constraint. In addition, the weights are set to zero locally in the algorithm for those inequality constraints that are currently inactive, i.e. those that, at the current estimation, satisfy the strict inequality version of (1.3).

These two objective functions do not carry equal weight, since our problem can be formulated as the maximization of L subject to the constraint $C = 0$. Thus we proceed through the iteration of two steps: one that decreases the value of the cost C, and another that increases the likelihood L while not increasing C, at least to leading order in the step-size.

1.2.1 Simulation of expected values and their evolution

At each stage of the map $y(x)$, one needs to evaluate expected values of the form

$$E_i = E(f_i; y(x)) = \int f_i(x)\rho(x)\ dx = \int f_i(x(y))\mu(y)\ dy. \tag{1.11}$$

where we have denoted by $x(y)$ the inverse of the map $y(x)$. It would appear natural to estimate these through Monte Carlo simulation:

$$E(f_i;y(x)) \approx \frac{1}{s}\sum_{l=1}^{s} f_i(x(y_l)),$$

where the y_l are s independent samples of the target μ, typically a normal distribution, easy to sample. Yet the problem with this prescription is the calculation of $x(y_l)$: these can be computed only if one has stored all the elementary maps z_k up to the current iteration and, moreover, these maps have a closed-form inverse. Even under this ideal scenario, the cost of the calculation would grow linearly with the iteration k, making it impractical. Instead, we propose to change measure to a yet unspecified distribution $\eta_k(y)$, and write

$$E(f_i;y(x)) = \int f_i(x(y))\frac{\mu(y)}{\eta_k(y)}\,\eta_k(y)\,dy \approx \frac{1}{s}\sum_{l=1}^{s} f_i(x(y_l))\frac{\mu(y_l)}{\eta_k(y_l)}, \tag{1.12}$$

where the y_l are now samples of $\eta_k(y)$. To avoid the problem described above of evaluating $x(y_l)$, we propose distributions $\eta_k(y)$ that evolve with the algorithm in such a way that the samples $x_l = x(y_l)$ are kept fixed. To this end, it is enough to propose an initial distribution $\eta_0(y)$ and sample it at the onset of the algorithm, when $y = x$, and then let the corresponding points y_l be carried passively by the maps. Then the corresponding $x(y_l)$ remain fixed at their original values, and the density $\eta_k(y)$ is updated at each step by division by the Jacobian of the map:

$$\eta_{k+1}(z_k(y)) = \frac{\eta_k(y)}{j_k(y)}. \tag{1.13}$$

In addition to evaluating the current expected values $E(f_i;y(x))$ at each step, we also need to estimate how they would evolve under a candidate map

$$z(y) = y + \varphi(y) \tag{1.14}$$

with Jacobian $j(y)$. The function $\varphi(y)$ is required to be small, yielding a map close to the identity. This justifies the linearization procedure described below.

Let us denote by

$$\rho^-(x) = J_y(x)\,\mu(y(x)), \tag{1.15}$$

the current estimate of ρ, given the cumulative effect of the maps up to step k, and by

$$\rho^+(x) = \kappa(y(x))\rho^-(x) \tag{1.16}$$

the estimate after the candidate step, where

$$\kappa(y) = j(y)\frac{\mu(y+\varphi(y))}{\mu(y)} \approx 1 + \frac{\nabla\cdot[\mu(y)\varphi(y)]}{\mu(y)}, \tag{1.17}$$

where $\nabla\cdot$ denotes the divergence operator, expanded up to linear terms in φ. Then

$$\begin{aligned}\Delta E_i(\varphi) &= E(f_i;z(y(x))) - E(f_i;y(x))\\ &= \int f_i(x)\left(\rho^+(x)-\rho^-(x)\right)\,dx = \int f_i(x)\left(\kappa(y(x))-1\right)\rho^-(x)\,dx\\ &\approx \int f_i(x(y))\nabla\cdot[\mu(y)\varphi(y)]\,dy = dE_i(\varphi).\end{aligned} \tag{1.18}$$

This can be estimated through the introduction of an auxiliary distribution $\eta_k(y)$ as above.

Next we describe the algorithm's two steps.

1.2.2 Reduction of the cost –the C-step

In order to reduce the cost C from (1.9), we propose a map of the form

$$z(y) = y + \varphi(y), \quad \varphi(y) = \sum_{h=1}^{n_h} \gamma_h \varphi_h(y), \tag{1.19}$$

where the $\varphi_h(y)$ are n_h suitably picked functions (more on this below), and compute the gradient and Hessian of C with respect to the γ_h:

$$G_h = \left.\frac{\delta C}{\delta \gamma_h}\right|_{\gamma=0} = \sum_i w_i \, (E_i - \bar{f}_i) \, dE_i(\varphi_h) \tag{1.20}$$

and

$$H_k^m = \left.\frac{\delta^2 C}{\delta \gamma_k \delta \gamma_m}\right|_{\gamma=0} \approx \sum_i w_i \, dE_i(\varphi_k) \, dE_i(\varphi_m). \tag{1.21}$$

In terms of the matrix A and vector b with entries

$$A_i^j = \sqrt{w_i} \, dE_i(\phi_j), \quad b_i = \sqrt{w_i} \, (E_i - \bar{f}_i), \tag{1.22}$$

we have that

$$G = A'b \quad \text{and} \quad H = A'A, \tag{1.23}$$

so Newton's method yields a vector γ satisfying the normal equations

$$A'A\gamma = -A'b. \tag{1.24}$$

Notice that, by invoking the linear approximation to $\kappa(y)$ in (1.17), our version of Newton's method uses a surrogate for the Hessian H, whose exact determination would require an expansion for $\kappa(y)$ involving quadratic terms in φ, namely

$$\begin{aligned} \kappa(y) = j(y)\frac{\mu(y+\varphi(y))}{\mu(y)} \approx 1 + \frac{\nabla\cdot[\mu(y)\varphi(y)]}{\mu(y)} + \sum_{j>i}\left[\frac{\delta\varphi_i}{\delta y_i}\frac{\delta\varphi_j}{\delta y_j} - \frac{\delta\varphi_i}{\delta y_j}\frac{\delta\varphi_j}{\delta y_i}\right] \\ + \frac{1}{\mu(y)}\left[\frac{1}{2}(\varphi\cdot\nabla)^2\mu(y) + (\nabla\cdot\varphi)(\varphi\cdot\mu(y))\right]. \end{aligned} \tag{1.25}$$

The justification for using a surrogate $H = A'A$ instead of the true Hessian, a common practice in least square problems [3], is the following:

1. The surrogate is positive definite, while the true Hessian need not be.
2. Less and simpler calculations are required to evaluate the surrogate.
3. At the minimum $C = 0$, the two Hessians agree: the true Hessian is given by

$$H_k^m = \sum_i w_i \, \frac{\delta C_i}{\delta \gamma_k}\frac{\delta C_i}{\delta \gamma_m} + \sum_i w_i \, C_i \, \frac{\delta^2 C_i}{\delta \gamma_k \delta \gamma_m}, \tag{1.26}$$

while the surrogate includes only the first of the two sums. But $C = 0$ implies that all the C_i's are zero, so the second sum vanishes.

Regarding the number n_h of parameters γ_h to use in this step, one is enough. We have carried the calculation for an arbitrary number n_h because some of the results above will be also used in the L-step described below, where two free parameters are required.

1.2.3 Increase of the likelihood function –the L-step

In this section we show how to carry out the second step in our program: increasing the log likelihood of the density ρ in (1.4) while not undoing the gains in cost of the previous step. From a geometric viewpoint, this amounts to seeking directions in the infinite dimensional space of maps φ such that the likelihood L increases while the cost C does not increase: if the gradients of L and C project positively on each other, the chosen direction must be tangent to the the *level sets* of the cost functional C given by (1.9).

We seek a map φ in the form

$$\varphi = \beta\left(\gamma_1\varphi_1 + \gamma_2\varphi_2\right),$$

where β is a free parameter available to ascend the log-likelihood, and the coefficients $\gamma_h, h = 1,2$ of the φ_h are chosen from considerations involving also the cost C. We first set β to one and compute the derivatives of C –as in (1.20)– and of L with respect to the gammas,

$$\nabla_\gamma C = \left.\begin{pmatrix} \frac{\partial C}{\partial \gamma_1} \\ \frac{\partial C}{\partial \gamma_2} \end{pmatrix}\right|_{\gamma=0}, \quad \nabla_\gamma L = \left.\begin{pmatrix} \frac{\partial L}{\partial \gamma_1} \\ \frac{\partial L}{\partial \gamma_2} \end{pmatrix}\right|_{\gamma=0}.$$

If $\nabla_\gamma L \cdot \nabla_\gamma C < 0$, we can adopt for the ascent of L its direction of maximal growth,

$$\begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = \frac{\nabla_\gamma L}{\|\nabla_\gamma L\|_2},$$

while simultaneously reducing C. Otherwise, we set $\gamma \perp \nabla_\gamma C$:

$$\gamma_1 \left.\frac{\partial C}{\partial \gamma_1}\right|_{\gamma=0} + \gamma_2 \left.\frac{\partial C}{\partial \gamma_1}\right|_{\gamma=0} = 0, \quad \|\gamma\|_2 = 1.$$

These two alternative determinations of γ are illustrated in figure 1.2.3. With γ thus fixed, we select β through second order descent:

$$\beta = -\frac{L_\beta}{L_{\beta\beta}},$$

possibly capped to prevent unreasonably large steps.

1.2.4 Duality

The algorithm seeks a transformation

$$x \to y(x)$$

such that the y's are distributed following an isotropic Gaussian:

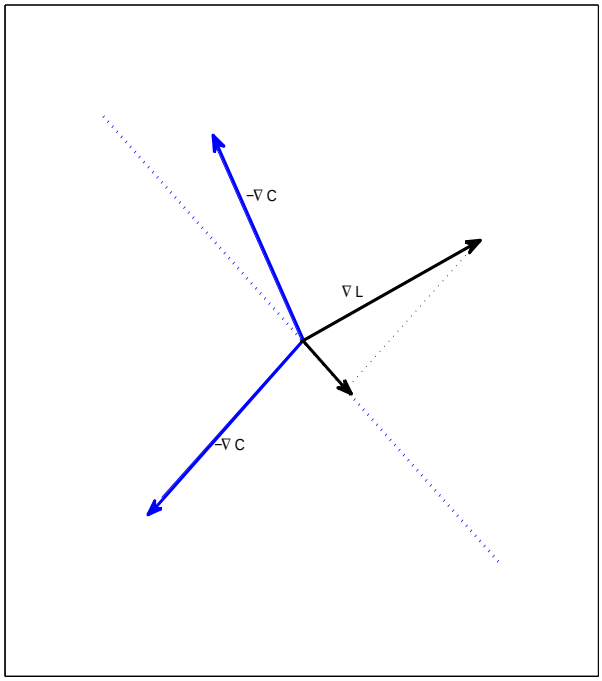

Fig. 1.1 Q plot exemplifying the two possible situations for the L-step: if the inner product of $\nabla_\gamma L$ and $-\nabla_\gamma C$ is positive, then $\gamma = \nabla_\gamma L$ is a permitted –and optimal– direction of ascent for L. Otherwise, the algorithm is constrained to ascend L along the line orthogonal to $\nabla_\gamma C$, not to deteriorate the current value of the cost C.

$$\mu(y) = \frac{1}{(2\pi)^{n/2}} e^{-\frac{1}{2}|y|^2}.$$

At each stage k of the algorithm, this provides an explicit formula for the estimated distribution in x-space:

$$\rho_k(x) = J^{y_k}(x)\, \mu(y_k(x)).$$

where $J^y(x)$ is the Jacobian of $y(x)$ evaluated at x. Then, at the next step, with $y_{k+1} = z(y_k)$ with Jacobian $j(y_k)$,

$$\rho_{k+1}(x) = j(y_k(x))\, J^{y_k}(x)\, \mu\left(z(y_k(x))\right),$$

so the log-likelihood L satisfies

$$L_{k+1} = \sum_j \log\left(\rho_{k+1}(x_j)\right) = \sum_j \log\left(J^{y_k}(x_j)\right) + \sum_j \log\left(j(y_j)\, \mu\left(z(y_j)\right)\right), \tag{1.27}$$

where $y_j = y_k(x_j)$, the current location of the transformed observations. Since the first summation on the right-hand side of (1.27) does not depend on the map $z(y)$, maximizing L_{k+1} over the candidate maps is equivalent to maximizing

$$\tilde{L}_{k+1} = \sum_j \log\left(j(y_j)\, \mu\left(z(y_j)\right)\right). \tag{1.28}$$

This maximization is indistinguishable from the first step ($k = 1$) of the algorithm, with the current y_j playing the role of the initial data x_j. Thus the algorithm can be formulated as a memory-less ascent

of the log-likelihood, which for the purpose of increasing L, forgets at every step the original values of the observations x_j. Alternatively, we can phrase this as a *duality* of the flow: as the random variable $y(x)$ acquires a normal distribution, the estimated density $\rho_k(x)$ converges to the real distribution $\rho(x)$ underlying the observations. One can introduce, in addition to the estimated density $\rho_k(x)$, the unknown density $\tilde{\rho}_k(y)$ evolving with the flow:

$$\tilde{\rho}_k(y) = \frac{\rho(x_k(y))}{J^{y_k}(x)}, \tag{1.29}$$

where $x_k(y)$ denotes the inverse of the map $y_k(x)$. It follows that

$$\tilde{\rho}_k(y) \to \mu(y) \Longleftrightarrow \rho_k(x) \to \rho(x) \ : \tag{1.30}$$

blindly moving the points y_j toward normality improves the density estimation for the x_j. Thus the maximization of $\tilde{L}_k$ in (1.28) can be thought of as a step toward normalizing the points y_j.

1.3 Fine tuning of the algorithm

1.3.1 Choice of the distribution $\eta(y)$

The distribution η is only used as a tool for computing the integrals in (1.12) and (1.18) through Monte Carlo via a change of sampling measure. It follows that one could choose different distributions η_i tailored to the individual functions $f_i(x)$ in the constraints. As described above, $\eta(y)$ is sampled only at the beginning of the algorithm, when $y(x) = x$, and then updated by the flow, through the expression in (1.13). In order that the change of sampling measure accurately reflects the original constraints in (1.2), the support of $\eta_i(x)$ must include the support of $f_i(x)\rho(x)$. One can move further in the direction of importance sampling, adopting an η_i for which the sampling frequency is high/low according to high/low values of $|f_i(x)|\rho(x)$.

One possible strategy for choosing $\eta_i(x)$ satisfying the requirements above is the following:

1. Compute the empirical mean μ and covariance matrix Σ of the datapoints x_j.
2. Define $\eta_0(x) = \mathcal{N}(\mu, \alpha\Sigma)$, where $\alpha > 1$ is a safety factor intended to guarantee that η_0 has significant sampling frequency over the full support of the density $\rho(x)$ underlying the samples x_j.
3. Define $\eta_i(x) = \frac{1}{Z} f(x)\eta_0(x)$, where $Z = \int f(x)\eta_0(x)\, dx$. This is a useful distribution for our purposes if it is easily sampled, as is the case for many financial instruments. Otherwise, if f has a well-defined maximum, we can rewrite it as $f(x) = e^{S(x)}$, and replace S by its quadratic approximation around its maximum, for which the resulting distribution is explicitly sampleable. Finally, in situations that cannot be handled in this or similar ways, one can simply adopt a uniform distribution $\eta_i = \eta_0$.

1.3.2 Choice of the functions φ_h

The building blocks of the algorithm are the functions $\varphi_h(y)$, the elementary maps of each step. There is much flexibility in the choice of a form for these maps, which must be guided by their effectiveness

to generate general maps by composition without over-resolving, and by their computational expense in evaluating the objective functions L and C and their derivatives.

A general strategy that we have found useful is to associate to each map φ_h a center y_h and a length scale or *bandwidth* α_h, writing

$$\varphi_h(y) = g\left(\frac{y-y_h}{\alpha_h}\right). \tag{1.31}$$

The scaling parameter α_h must depend on the selected node y_h, not to over-fit or under-resolve the data: in areas scarcely populated, α_h must be larger than in areas with high density, so that the transformation affects the likelihood of a similar number of observations. More precisely, given a number n_p of points that one would like to lie within a ball or radius α_h around y_h, α_h is given to leading order by

$$\alpha_h = \left(\frac{n_p\ \tilde{\rho}_k(y_h)}{m\ \Omega_n}\right)^{\frac{1}{n}}. \tag{1.32}$$

Here Ω_n is the volume of the unit ball in R^n. Since $\tilde{\rho}_k(y_h)$ is not known, however, we may replace it by its target normal distribution, which yields

$$\alpha_h = (2\pi)^{\frac{1}{2}}\left(\Omega_n^{-1}\frac{n_p}{m}\right)^{\frac{1}{n}} e^{\frac{\|y_h\|^2}{2n}}. \tag{1.33}$$

If deemed necessary, one can still use (1.32) in a second pass of the algorithm, estimating $\tilde{\rho}_k(y_h)$ through $\frac{\bar{\rho}(x_h)}{J_k(x_h)}$, where x_h is defined by the condition that $y_h = y_k(x_h)$, and $\bar{\rho}(x_h)$ is its estimated density from the first pass.

Regarding the selection of the nodes y_h, we alternate between two methodologies: to pick them at random among the current normalized observations y_j, and to sample them from the target distribution $\mu(y)$ (see [2] for the rationale behind this alternating procedure). In the second pass of the algorithm as described above, only the first methodology can be used, since otherwise to find x_h one would need to invert all the maps z_k performed so far.

The functional form of the maps, $g(y)$, is almost completely unconstrained in one dimension; in the one-dimensional examples in section 1.4.1, we have used

$$g(y) = \tan^{-1}(y). \tag{1.34}$$

In higher dimensions, a practical choice is given by radial expansions of the form

$$g(y) = f(|y|)y, \tag{1.35}$$

with a scalar function $f(r)$ that localizes the action of the map to a neighborhood of y_h. This can range from a slowly decaying function, such as

$$f(r) = \frac{\mathrm{erf(r)}}{r}. \tag{1.36}$$

to one with compact support, such as

$$f(r) = (1-r)^2 \quad \text{for } r < 1, \quad f(r) = 0 \quad \text{otherwise.} \tag{1.37}$$

Yet a practical constraint arises in multidimensional scenarios when the number n_h of elementary maps in one step is larger than one, as needs be in the L-step of the algorithm, that has $n_h = 2$: the Jacobian determinant $j(y)$ of the map $\varphi(y)$ in (1.19) is a nonlinear function that couples all the γ_h, which makes the procedure more computational intensive. To avoid this, one may choose functions

$f(r)$ with compact support –the one proposed in (1.37) in the examples below– and pick the nodes y_h and bandwidths α_h so that the support of the various φ_h do not overlap, thus making the Jacobian of the map and its derivatives as easy to compute as when $n_h = 1$. Notice though that the nonlinearity of the Jacobian does not manifest itself at the time of computing $L_\beta\big|_{\beta=0}$, since here straightforward superposition applies. As for $L_{\beta\beta}\big|_{\beta=0}$, a surrogate can be used as with the Hessian of the cost, involving the Jacobian of each map ϕ_h separately. Hence it is only at the time of updating the map that the nonlinearity of the Jacobian comes into play, so the added computational cost from using overlapping maps ϕ_h is not so big, at least in comparatively low dimensional settings.

One further required property of the maps is that they should "see" the constraints, at least for the C-step of the algorithm, else they cannot decrease the cost. One easy way to satisfy this requirement is to choose the centers y_h of the spheres in such a way that $f_i(x(y_h)) \neq 0$ at least for one value of i.

1.3.3 Choice of the weights w_i

The cost function C in (1.9) depends on the weights w_i assigned to each individual constraint. These have two roles: balancing the generally different intrinsic variability of the various constraints, and enforcing the level of significance that the user attaches to each. A simple recipe appropriate to the first role is to make the weights w_i inversely proportional to the empirical variance of the corresponding $f_i(x)$:

$$w_i \propto \left[\sum_j \left(f_i(x_j) - \bar{f}_i\right)^2\right]^{-1}, \tag{1.38}$$

with the proportionality constant fixed so that $\sum_i w_i = 1$. For certain particular functions f_i, such as the indicator functions of (1.1), the empirical variance above can yield a zero value and corresponding infinite weight. This might be addressed by weighting in also the empirical variance of f_i over the sample points of η_i used for Monte Carlo simulation. As for the more subjective second role, it is up to the user to multiply each of the weights above by an individual factor that quantifies the relative importance of enforcing the corresponding constraint.

1.3.4 Inequality constraints

The procedure described so far applies almost without change to handle inequality constraints of the form (1.3). The only extra ingredient is the determination, at the onset of each step, of whether each inequality constraint is or not already satisfied. If it is not, the corresponding constraint can be treated as an equality, since the local goal is to reduce it toward zero. Otherwise, the constraint is deemed currently inactive and removed from the cost C (for instance, simply but not soundly from the computational viewpoint, by setting its weight w_i to zero.)

1.4 Examples

This section illustrates through a number of numerical examples the effect of imposing various kinds of constraints on density estimation. All the examples presented are synthetic, though motivated by real applications. Though the procedure developed in this article applies to arbitrarily large dimen-

sions, the examples are low-dimensional, to facilitate visualization. In a similar tone, the densities proposed all have simple structure and analytical forms, so that one can concentrate on the new features brought about by the imposition of constraints.

Besides showing the algorithm at work, the main message that these examples intend to convey is the versatility of constrained density estimation. This goes far beyond the applications that initially motivated its development, such as enforcing the compatibility of a pricing measure with the option prices available. In the modeler's hand, it can be used for tasks as diverse as enforcing symmetries, reducing oscillations and excluding forbidden areas of phase space.

In this first paper on constrained density estimation, we have not attempted to fully optimize the algorithm's implementation. In particular, the choice of the distribution η_i for the Monte Carlo simulation of the constraints in all examples but the last, is the simplest $\eta_i = \eta_0$, where η_0 is the Gaussian introduced in subsection 1.3.1, with safety factor $\alpha = 4$.

1.4.1 One dimensional examples

We start with some simple one-dimensional examples: the exponential distribution and the uniform distribution in the interval $[0,1]$. We contrast the results of unconstrained density estimation with the refinements obtained by the imposition of a variety of constraints.

In all the one-dimensional examples presented here, we have used elementary maps as in (1.31), with $g(y) = \tan^{-1}(y)$.

An exponential distribution

The exponential distribution

$$\rho(x) = e^{-x} \quad \text{for } x \geq 0, \quad \rho(x) = 0 \quad \text{otherwise} \tag{1.39}$$

represents a severe test for the algorithm, since mapping it into a Gaussian requires a singular transformation, that maps $x = 0$ to $y = -\infty$. We take a random sample of 1000 observations x_j from (1.39), and use it to estimate $\rho(x)$. Figures 1.2 and 1.3 display the results of applying the procedure to the data without imposing any constraint. The plots in figure 1.2 represent various densities: the true exponential underlying the data, the initial Gaussian estimation, with the empirical mean and variance of the data, and the estimated density after the first and second pass of the algorithm, the latter using the estimation from the former to refine the calculation of the bandwidths at each step through (1.32). The plots in figure 1.3 are histograms of the data points, before and after the normalization performed by the algorithm.

Generally, the results in figure 1.2 exemplify the power of the density-estimation component of the algorithm, whose non-parametric nature allows it to accurately represent distributions that are very far from the initial Gaussian guess. The main discrepancy between the true and estimated density lies in the mollification in the latter of the discontinuity at $x = 0$. The dual manifestation of this mollification is reflected in the histogram on the right of figure 1.3, where the normalization process is clearly incomplete: a thorough normalization would need to smear the discontinuity on the left into the whole negative semi-axis. One can decrease this mollification by a reduction of the algorithm's bandwidth, but this would lead to over-resolution elsewhere.

The mollification of the discontinuity at $x = 0$ results is assigning non-zero probability density to values of x smaller than zero. Even though our exponential distribution here is a synthetic construct,

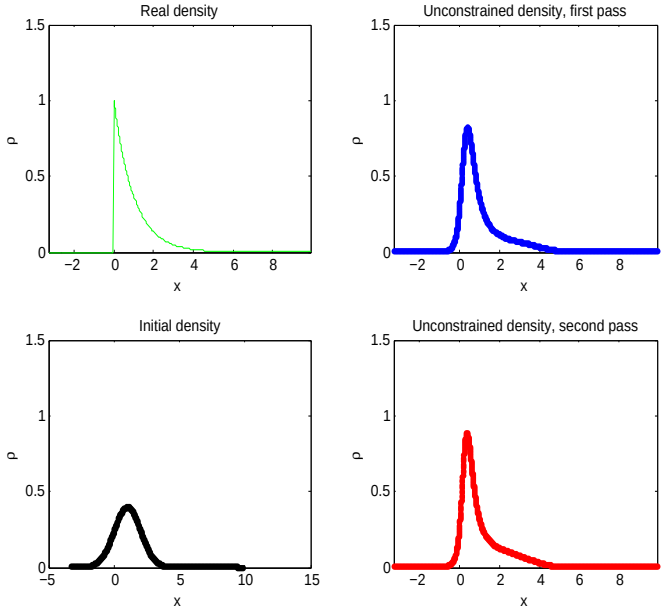

Fig. 1.2 Unconstrained density estimation of an exponential distribution. On the upper-left panel, the exact exponential density underlying the data. The other three panels display the density estimated at various stages of the algorithm. On the lower-left, the result of a linear preconditioning step, that maps the empirical mean to zero and rescales x so that the empirical variance is one. The resulting estimate is a Gaussian distribution with the mean and variance of the data. On the upper-right panel, the density estimated after the first pass of the algorithm, which computes the bandwidth α at each step using (1.33). On the lower-right, the result of the second pass, with α from (1.32) with the density estimated from the first pass.

it seems fair to assume that, in real applications, it would be associated with a variable x that cannot adopt negative values, such as a waiting time. If this constraint $x \geq 0$ were known a priori, we could use the procedure of this article to impose it through the expected value of the indicator function of the negative axis:

$$\int_{-\infty}^{0} \rho(x)\, dx = 0. \tag{1.40}$$

The results of imposing this constraint are displayed in figures 1.4, 1.5 and 1.6. One can see a much sharper discontinuity at $x = 0$, with almost no mass in the negative semi-axis, and a correspondingly more thorough normalization of the data.

A uniform distribution

As a second one-dimensional example, we consider the uniform distribution

$$\rho(x) = 1 \quad \text{for } 0 \leq x \leq 1, \quad \rho(x) = 0 \quad \text{elsewhere.} \tag{1.41}$$

We draw again a sample of 1000 observations and perform first an unconstrained density estimation. The results are displayed in figure 1.7. As before, the first pass yields reasonable results, yet greatly

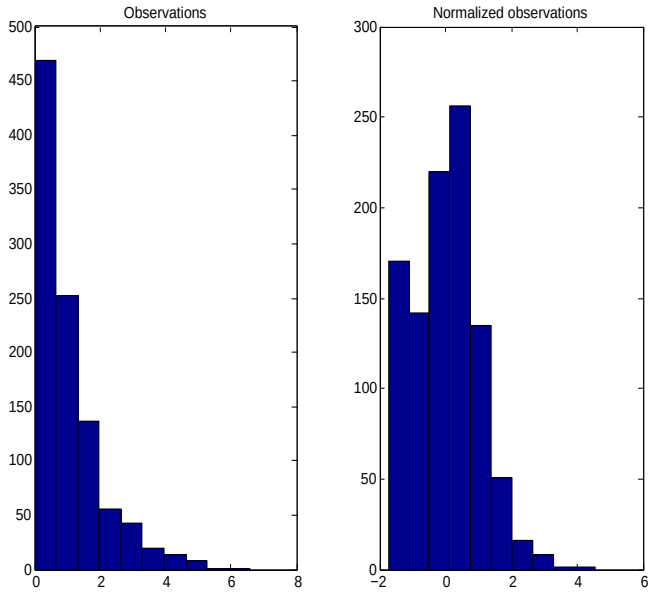

Fig. 1.3 Unconstrained density estimation of an exponential distribution. Histograms of the data-points: on the left, the original sample; on the right, the sample after normalization. The incomplete normalization on the left of the histogram is the dual manifestation of the mollification of the discontinuity at $x = 0$ in the estimated density of figure 1.2.

mollifies the discontinuities at $x = 0$ and $x = 1$. The second pass reduces this mollification, but at the expense of overshooting at $x = 0^+$ and $x = 1^-$, in a pattern reminiscent of Gibb's phenomenon.

To remedy these deficiencies using external information, we consider the scenario of a financial application, with x representing the logarithm of the return of an asset at some future time. We assume that, by regulation, this return has strict upper and lower bounds, yielding the constraints

$$0 \leq x \leq 1. \tag{1.42}$$

Moreover, we know the prices of 50 call and put options:

$$E_i = \int_0^1 \max\left(e^x - k_i, 0\right) dx \quad \text{or} \quad \int_0^1 \max\left(k_i - e^x, 0\right)\big) dx,$$

for strike prices $k_i = e^{\Delta x} \ldots e^{1-\Delta x}$, with $\Delta x = 1/51$, arbitrarily assigned to call and put options for i odd and even respectively. We compute the E_i explicitly and provide it to the programs as constraints, additional to the bounds in (1.42). The results are displayed in figures 1.8 and 1.9.

The estimated distribution after the second pass is now much more accurate. The remaining wiggles are in fact a reflection of fluctuations in the actual data, as seen on the histogram on the left of figure 1.9. One can reduce them further by adopting a coarser resolution, i.e. a larger bandwidth, but at the expense of mollifying the two discontinuities at $x = 0$ and $x = 1$.

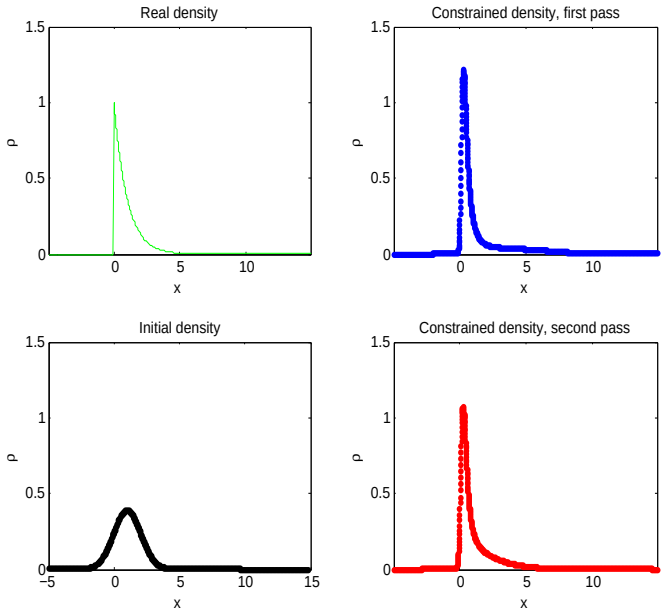

Fig. 1.4 Same as figure 1.2, but with the imposed constraint $x \geq 0$.

1.4.2 Multidimensional examples

Though the procedure is general, we display here only two-dimensional examples of constrained multidimensional density estimation: a Gaussian mixture required to satisfy a simple symmetry, and a two-dimensional Student t constrained by the prices of some liquid options.

In both cases, we have adopted as elementary maps the radial expansions in (1.35), with $f(r)$ given by (1.36).

A Gaussian mixture

As a first example, we consider the two-dimensional, two-component Gaussian mixture

$$\rho(x) = \sum_{k=1}^{2} \gamma_k N(x, \mu_k, \Sigma_k), \tag{1.43}$$

with weights

$$\gamma_1 = \gamma_2 = \frac{1}{2},$$

centers

$$\mu_1 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \quad \mu_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

and isotropic covariance matrices

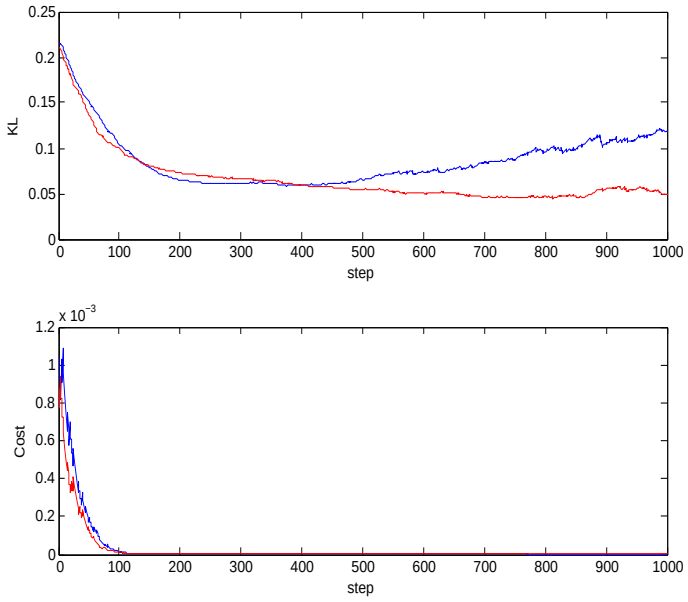

Fig. 1.5 Diagnostics for the estimation of an exponential distribution from a sample and the imposed constraint that $x > 0$. In blue the first pass, in red the second. On the top panel, the evolution of the Kullback-Leibler divergence between the exact density and the estimated one. On the lower panel, the evolution of the cost C associated with the non-satisfaction of the constraint.

$$\Sigma_1 = \Sigma_2 = \begin{pmatrix} 0.2 & 0 \\ 0 & 0.2 \end{pmatrix}.$$

The results of the corresponding unconstrained density estimation are displayed in figure 1.10.

The estimated density is fine qualitatively, again displaying the versatility of the non-parametric approach through iterated maps, that can capture quite arbitrary distributions without external input other than sample points. Yet one feature missed is the symmetry of the distribution with respect to the y axis: due to a random sampling fluctuation, the Gaussian component on the left had a larger number of observations than the one on the right; as a consequence, the corresponding density has a higher peak. If we had known of the right-left symmetry from first principles of the problem in hand, we could had imposed it at various levels of sophistication and detail, the simplest being the balancing constraint

$$E(x_1) = 0. \tag{1.44}$$

Figures 1.14, 1.12 and 1.13 display the results of imposing this single constraint: the estimated density is far more symmetric than in the unconstrained estimation.

A Spread Option

In our last example, we consider the problem of estimating the risk neutral joint density of two assets at a single maturity, assuming that options are liquidly traded on each of the assets, but illiquidly traded on the spread. The example thereof mentioned in the introduction is a crack spread option. Here, for the purpose of illustration we assume the following:

- Each asset $S_{1,2}$ has for the maturity considered options trading with 11 different strikes.
- The spread option has options trading with 6 different strikes.

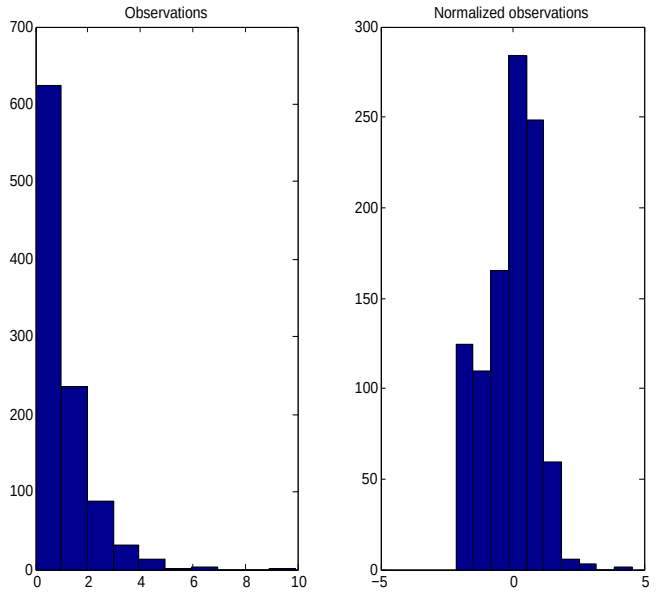

Fig. 1.6 Same as figure 1.3, but with the imposed constraint $x \geq 0$. Notice the more complete normalization on the left of the histogram.

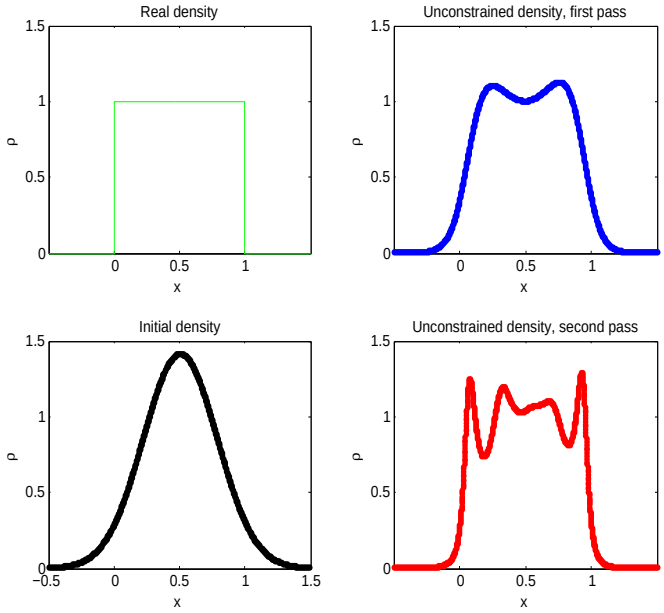

Fig. 1.7 Same as figure 1.2, but for a uniform distribution.

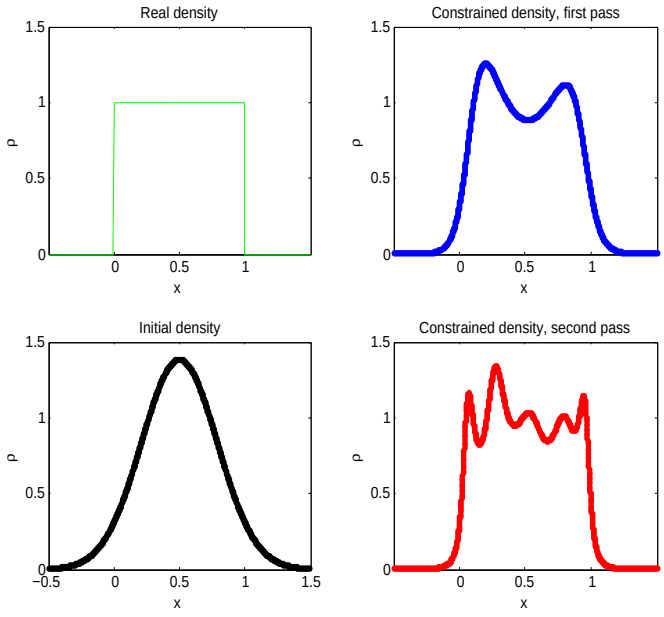

Fig. 1.8 Density estimation of a uniform distribution, constrained by lower and upper bounds and by 50 call and put option prices. On the upper-left panel, the exact uniform density underlying the data. The other three panels display the density estimated at various stages of the algorithm. On the lower-left, the Gaussian resulting from the linear preconditioning step. On the upper-right panel, the density estimated after the first pass of the algorithm, with the bandwidth α at each step computed from (1.33). On the lower-right, the result of the second pass, with α from (1.32) with the density estimated from the first pass.

- We generate option prices for each underlying by simulation, assuming that the true joint distribution and marginals come from a multivariate student-T distribution with 5 degrees of freedom and correlation matrix $C = \begin{pmatrix} 1 & .6 \\ .6 & 1 \end{pmatrix}$.

Not surprisingly, since the student-t distributions have heavy tails the call prices exhibit a distinct smile, as illustrated in Figure 1.15.

We assume the spot prices are $S_1 = 10, S_2 = 12$. To determine the option prices, we simulate from this known distribution with high accuracy, using 100,000 samples from a bivariate student-t with 5 degrees of freedom. We then extract an independent sample from the same distribution containing only 1,000 pairs of data points. These, together with the precise option prices mentioned above, are the data provided to our algorithm to generate an estimate of the probability distribution. Figure 1.15 shows the true density of the data points in the North-West corner and the estimated density in the South-East corner. By observing the distribution of the points in our sample, it is clear that the Monte-Carlo points used to evaluate the cost associated with the constraints should be drawn from a distribution with fatter tails than the normal distribution used in our previous examples. Thus, in this example, we generate Monte-Carlo points, using a distribution that is not far from normal, but has fatter tails, namely a bivariate student-t distribution with 15 degrees of freedom.

Figure 1.18 shows the evolution of the Kullback-Leibler divergence and the evolution of the cost with the number of steps in the algorithm, while figure 1.19 shows the initial and final Monte Carlo points.

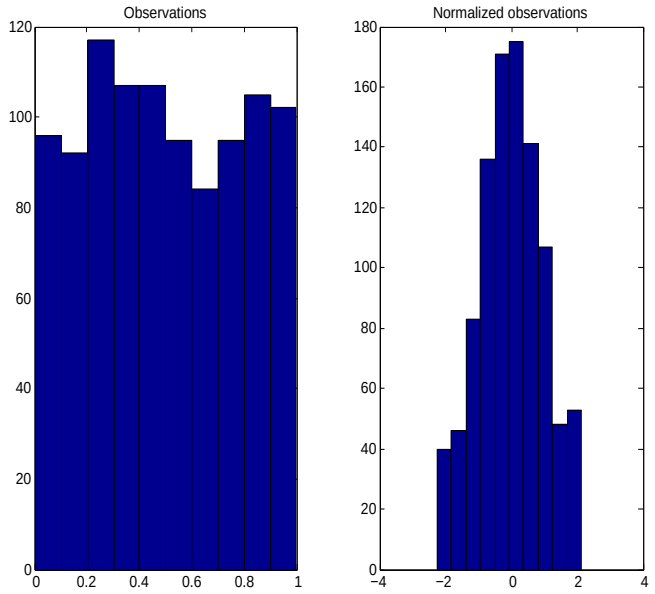

Fig. 1.9 Density estimation of a uniform distribution constrained by bounds and option prices. Histograms of the datapoints: on the left, the original sample; on the right, the sample after normalization. An effect of the finite size of the sample is the appearance of fluctuations in the histogram of the observations, that the algorithm interprets as true density fluctuations in its estimation on the lower right panel of figure 1.8. This over-resolution can be corrected by increasing the algorithm's bandwidth, but at the expense of mollifying further the discontinuities at the two ends of the support of $\rho(x)$.

In addition we have performed the following test to see how well the algorithm can use the information inherent in the 1,000 observations and in the option prices to estimate an additional option price not provided as a constraint. In the experiment, we provide the algorithm with all of the call option prices for S_1 and for S_2 as well as the spread option prices, in the second column of Table 1, *with the exception of* the spread option with strike 3.3, which it is asked to estimate. The at the money value of the spread corresponds to $K = 2$ and the spread option prices decrease rapidly for larger strikes.

Table 1 compares the performance of the algorithm, subject to different prescriptions of the weights attached to each constraint in the global cost C. In column 4, the weights are chosen according to the prescription (3.8) from section 3.3. In column 5 (moderate weights), we evenly weight the 6 spread options that are used as constraints in the program (in addition to the 11 constraints on each marginal), in such a way that the total weight assigned the spreads equals $\frac{1}{2}$. In column 6, we heavily weight the spread option constraints by assigning a weight $\frac{3}{4}$ to the spread option constraints and only $\frac{1}{4}$ to the rest. In this example, moderate weighting performs better on the extra option with strike 3.3, giving an almost perfect match.

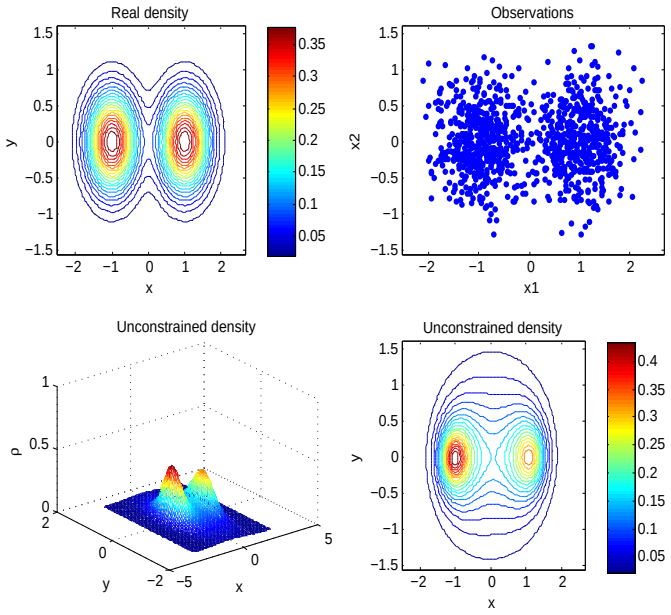

Fig. 1.10 Two-dimensional unconstrained density estimation: equidistributed mixture of two isotropic Gaussians. On the upper-left panel, contour lines of the exact uniform density underlying the data. On its right, the sample points used for density estimation. On the lower panels, the estimated density in perspective and contour lines.

1.5 Conclusions

This article introduces a new methodology for density estimation with constraints on the expected value of a given set of functionals. The methodology is based on normalizing the data through the composition of simple near-identity maps, driven by the ascent of the likelihood of the estimated density and the descent of a cost associated with the non-satisfaction of the constraints. The constraints are simulated through an important-sampling Monte Carlo technique with sample points that follow the flow defined by the maps of the algorithm.

The power and versatility of the methodology is illustrated through a few synthetic low-dimensional examples. Many further refinements are possible; some are hinted at in the text. The applicability of the methodology is wide, including many financial and bio-medical applications.

References

1. Tabak, E. and Vanden-Eijnden, E., "Density estimation by dual ascent of the log-likelihood", *Comm. Math. Sci*.**8**, 217-233, 2010.
2. Tabak, E. and Turner, C., "A family of non-parametric density estimation algorithms", to appear in *CPAM*, 2012.
3. Nocedal, J. and Wright, S. J., **Numerical optimization**, *Springer Series in Operations Research and Financial Engineering*, Springer, New York, 2006.

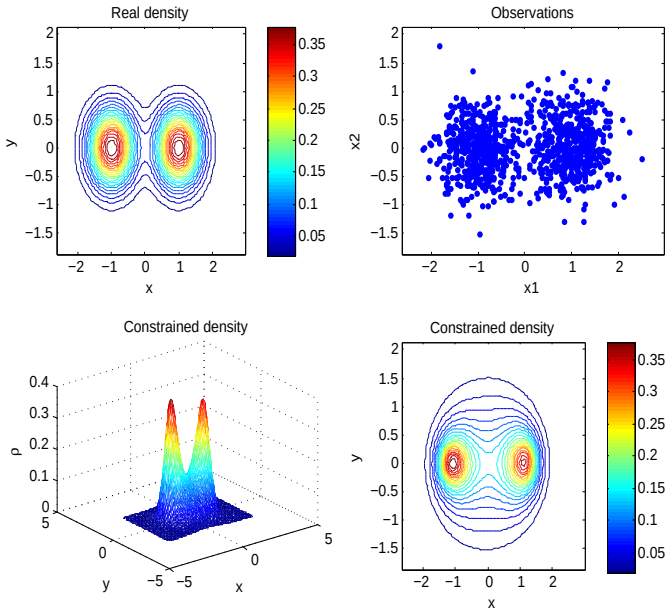

Fig. 1.11 Same as figure 1.10, but imposing the constraint (1.44). Now the left-right symmetry of the distribution is much more evident in the estimated density.

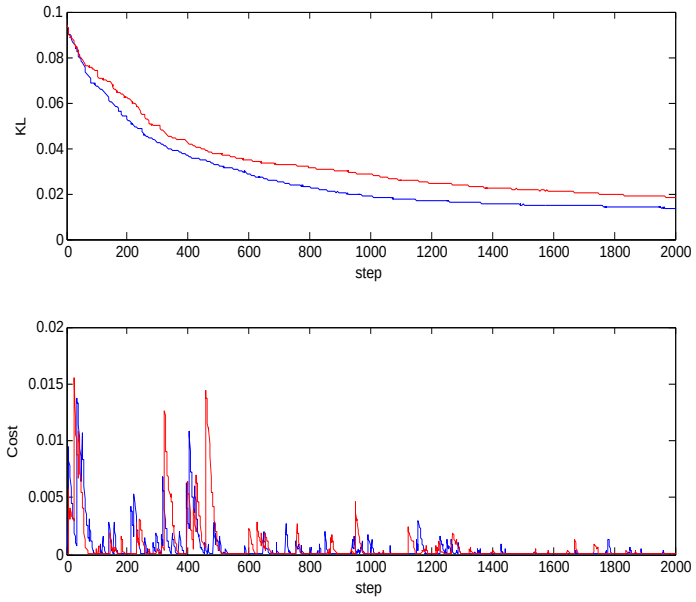

Fig. 1.12 Diagnostics for the estimation of the Gaussian mixture in (1.43) subject to the constraint in (1.44). Blue stands for the first pass, red for the second. On the top panel, the evolution of the Kullback-Leibler divergence between the exact density and the estimated one. On the lower panel, the evolution of the cost C associated with the non-satisfaction of the constraint.

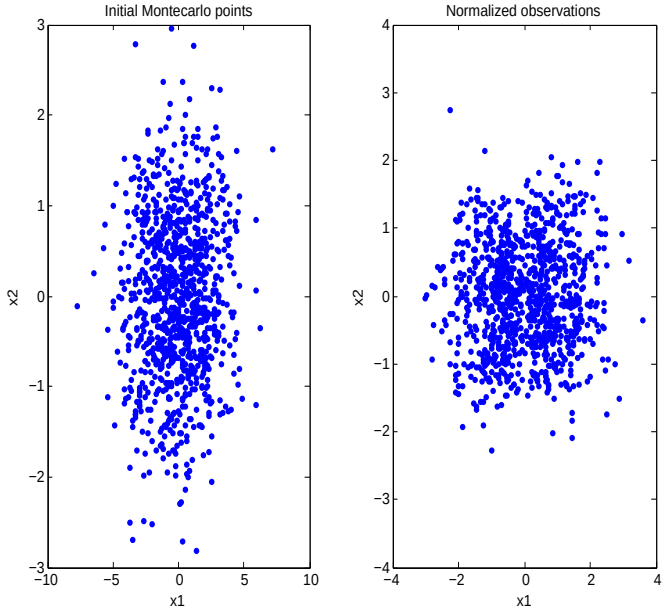

Fig. 1.13 Two plots illustrating details of the estimation of the Gaussian mixture in (1.43) subject to the constraint in (1.44). On the left, the initial sample of $\eta(x)$, used for the Monte Carlo simulation of the constraint. In this case, we have adopted the simplest choice of a Gaussian $\eta(x)$ with the empirical mean of the data points and four times its empirical covariance matrix. On the right, the observations on the upper-right panel of figure 1.14, after the normalization performed by the procedure.

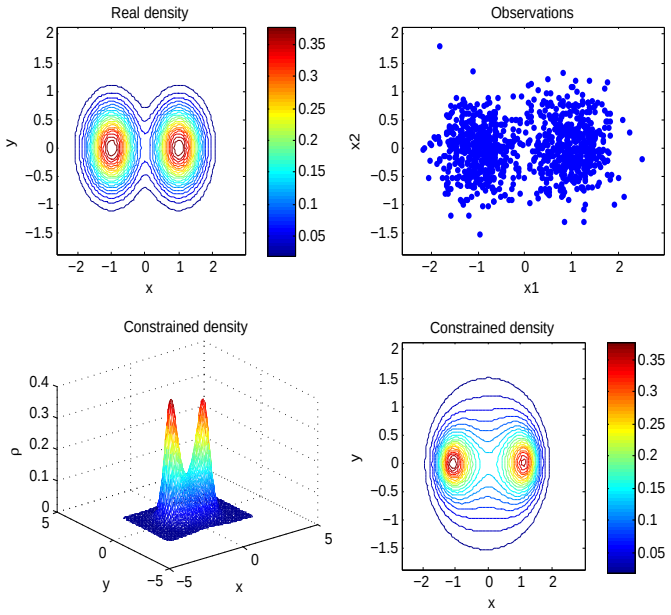

Fig. 1.14 Same as figure 1.10, but imposing the constraint (1.44). Now the left-right symmetry of the distribution is much more evident in the estimated density.

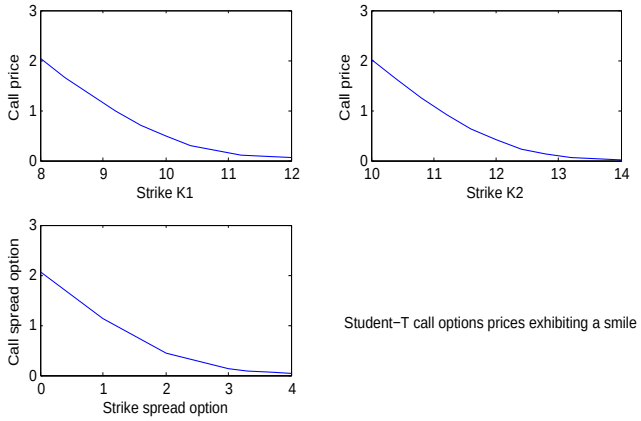

Fig. 1.15 Smile profiles produced by a bivariate fat-tailed student-T distribution with 5 degrees of freedom and mean $[10, 12]$

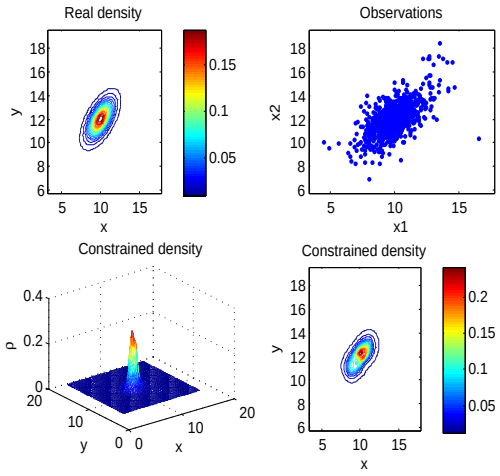

Fig. 1.16 The plot in the upper left hand corner depicts the true distribution, a bivariate student-T with 5 degrees of freedom. There are 26 constraints, consisting of call options on 11 strikes on each underlying and 6 on the spread. The lower right hand corner shows the estimated constrained density after the first pass.

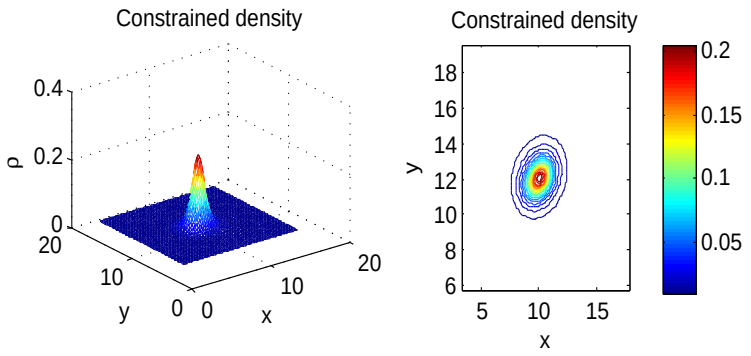

Fig. 1.17 The estimated constrained density after the second pass, using the weights indicated in (3.8), is found to be quite similar to the true density, in the upper left hand corner of figure 1.16.

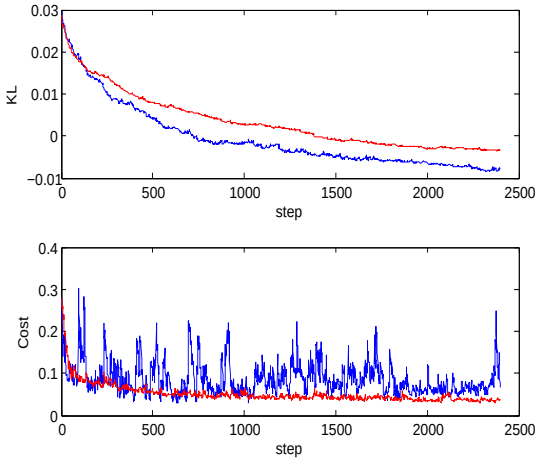

Fig. 1.18 The evolution of the Kullback Leibler divergence and of the cost as a function of the number of iterations

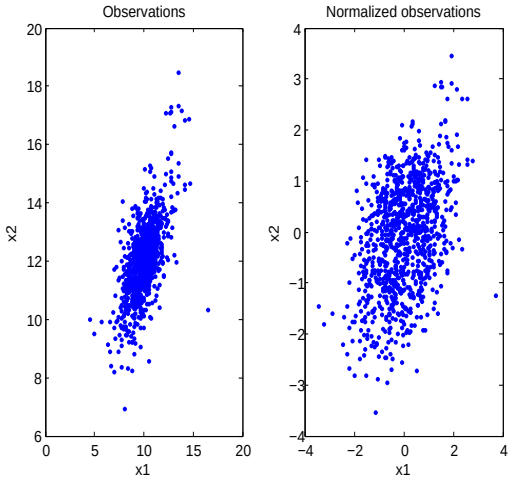

Fig. 1.19 The initial and final Monte-Carlo points chosen from a student-t distribution with 15 degrees of freedom.

Strike	Prices:				
	True	Emp.	Est.	with	weights
K1			(38)	Moder	Extreme
8	2.0469	2.0312	2.0379	2.0507	2.024
8.4	1.6734	1.6567	1.6599	1.6735	1.6544
8.8	1.318	1.3021	1.2962	1.3114	1.2938
9.2	0.9909	0.9742	0.9578	0.9825	0.9568
9.6	0.7058	0.694	0.6611	0.7005	0.6607
10	0.4756	0.4702	0.4306	0.4754	0.4295
10.4	0.3051	0.3016	0.2725	0.3122	0.2698
10.8	0.1898	0.1865	0.171	0.1954	0.1648
11.2	0.1165	0.1174	0.1016	0.1166	0.1016
11.6	0.0718	0.0753	0.0563	0.066	0.0608
12	0.045	0.0489	0.0284	0.0373	0.0383
K2					
10	2.009	2.0886	2.0055	2.0219	2.0139
10.4	1.6241	1.7139	1.6307	1.6456	1.6466
10.8	1.2572	1.3557	1.2734	1.2816	1.2922
11.2	0.9214	1.0237	0.944	0.943	0.9603
11.6	0.6316	0.7366	0.6545	0.6443	0.6672
12	0.4001	0.4974	0.4214	0.4073	0.4344
12.4	0.2321	0.3198	0.2526	0.2429	0.2649
12.8	0.122	0.201	0.1453	0.1401	0.1497
13.2	0.0578	0.1252	0.081	0.0784	0.0804
13.6	0.0245	0.0805	0.0406	0.0437	0.0438
14	0.0092	0.0554	0.0206	0.0242	0.0232
Ks					
0	2.0253	2.0748	0.0206	1.9511	1.9422
1	1.1093	1.1439	2.0196	1.1143	1.0979
2	0.4199	0.4337	1.1607	0.4662	0.4573
3	0.1123	0.1116	0.4913	0.1253	0.1384
3.3	0.0744	0.072	0.1647	0.0748	0.9
3.7	0.0438	0.0399	0.0821	0.0345	0.046
4	0.0301	0.0245	0.0613	0.0169	0.0265

Table 1.1 Option prices for the strikes in column 1. The true prices based in column 2, the empirical ones for the sample of 1,000 points in column 3, and three estimated prices in the last three columns, corresponding to three different ways to choose the weights in the cost functional, that place an increasing amount of emphasis on enforcing the spread option constraints. The spread option with strike $Ks = 3.3$ was not supplied to the algorithm as a constraint to enforce, but was instead calculated from the estimated densities.