Department of Statistics and Actuarial Science
University of Hong Kong

Tutorial 10: Optimization in practice and Numerical Integration

Xuran Meng

November 20, 2022

Overview

- Logistic Regression
  - Problem formulation
  - Newton–Raphson method
  - Practice
- MLE Estimator
  - Problem formulation
  - Newton–Raphson method
  - Practice
- Gaussian Quadrature
  - The basics
  - Connections with the Importance Sampling
  - Practice
- Quasi Monte Carlo
  - Problem formulation
  - Theories
  - Practice
- References

Logistic Regression

Logistic Regression

Problem formulation - the definition

- Consider the logistic model

$$\mathbb{P}(Z = 1|\boldsymbol{x}) = \mathrm{expit}(\boldsymbol{x}^T\boldsymbol{\theta}), \mathrm{expit}(x) = \frac{\exp(x)}{1 + \exp(x)}.$$

In other word,

$$\mathrm{logit}(\mathbb{P}(Z = 1|\boldsymbol{x})) = \boldsymbol{x}^T\boldsymbol{\theta}, \mathrm{logit}(p) = \log(\frac{p}{1-p}).$$

- If you compare it to the probit model

$$\mathbb{P}(Z = 1|\boldsymbol{x}) = \Phi(\boldsymbol{x}^T\boldsymbol{\theta}),$$

where Φ is the cdf of the standard normal distribution.

Logistic Regression

Problem formulation - the likelihood

- Given data $\{(z_1, \boldsymbol{x}'_1), \ldots, (z_n, \boldsymbol{x}'_n)\}$, if there are some same $\boldsymbol{x}'_i$, data can be summarized as $\{(y_1, n_1, \boldsymbol{x}_1), \ldots, (y_m, n_m, \boldsymbol{x}_m)\}$.
- The log-likelihood of θ is

$$
\begin{aligned}
l(\theta) &= \log\Big(\prod_{i=1}^{n} \mathbb{P}(Z = z_i | \boldsymbol{x}'_i)\Big) \\
&= \sum_{i=1}^{n} \log\big(\mathbb{P}(Z = 1 | \boldsymbol{x}'_i)^{z_i} \mathbb{P}(Z = 0 | \boldsymbol{x}'_i)^{1-z_i}\big) \\
&= \sum_{i=1}^{n} z_i \log\Big(\frac{\exp(\boldsymbol{x}'^{T}_i \theta)}{1 + \exp(\boldsymbol{x}'^{T}_i \theta)}\Big) + (1 - z_i) \log\Big(\frac{1}{1 + \exp(\boldsymbol{x}'^{T}_i \theta)}\Big) \\
&= \sum_{i=1}^{n} z_i (\boldsymbol{x}'^{T}_i \theta) - \log(1 + \exp(\boldsymbol{x}'^{T}_i \theta)) \\
&= \sum_{j=1}^{m} y_j (\boldsymbol{x}^{T}_j \theta) - n_j \log(1 + \exp(\boldsymbol{x}^{T}_j \theta)).
\end{aligned}
$$

Logistic Regression

Newton–Raphson method - the score

- Take the gradient of the log-likelihood

$$
\begin{aligned}
\nabla l(\theta) &= \sum_{j=1}^{m} \nabla y_j(\boldsymbol{x}_j^T\theta) - n_j \nabla \log(1 + \exp(\boldsymbol{x}_j^T\theta)) \\
&= \sum_{j=1}^{m} y_j \boldsymbol{x}_j - n_j \frac{\exp(\boldsymbol{x}_j^T\theta)\boldsymbol{x}_j}{1 + \exp(\boldsymbol{x}_j^T\theta)} \\
&= \sum_{j=1}^{m} \boldsymbol{x}_j (y_j - n_j \frac{\exp(\boldsymbol{x}_j^T\theta)}{1 + \exp(\boldsymbol{x}_j^T\theta)}) \\
&= \sum_{j=1}^{m} \boldsymbol{x}_j (y_j - n_j p_j), p_j = \frac{\exp(\boldsymbol{x}_j^T\theta)}{1 + \exp(\boldsymbol{x}_j^T\theta)}.
\end{aligned}
$$

Logistic Regression

Newton–Raphson method - the observed information

- Take the negative Hessian of the log-likelihood

$$
\begin{aligned}
-\nabla^2 l(\theta) &= -\nabla \sum_{j=1}^{m} \boldsymbol{x}_j \left(y_j - n_j \frac{\exp(\boldsymbol{x}_j^T\theta)}{1+\exp(\boldsymbol{x}_j^T\theta)}\right) \\
&= \sum_{j=1}^{m} \boldsymbol{x}_j n_j \nabla \frac{\exp(\boldsymbol{x}_j^T\theta)}{1+\exp(\boldsymbol{x}_j^T\theta)} \\
&= \sum_{j=1}^{m} \boldsymbol{x}_j n_j \frac{\exp(\boldsymbol{x}_j^T\theta)(1+\exp(\boldsymbol{x}_j^T\theta)) - \exp(\boldsymbol{x}_j^T\theta)^2}{(1+\exp(\boldsymbol{x}_j^T\theta))^2} \boldsymbol{x}_j^T \\
&= \sum_{j=1}^{m} \boldsymbol{x}_j \boldsymbol{x}_j^T n_j \frac{\exp(\boldsymbol{x}_j^T\theta)}{(1+\exp(\boldsymbol{x}_j^T\theta))^2} \\
&= \sum_{j=1}^{m} \boldsymbol{x}_j \boldsymbol{x}_j^T n_j p_j (1-p_j).
\end{aligned}
$$

Logistic Regression

Newton–Raphson method - the algorithm

- The Newton–Raphson method is

$$\begin{aligned}\boldsymbol{\theta}^{(t+1)} &= \boldsymbol{\theta}^{(t)} - (\nabla^2 l(\boldsymbol{\theta}^{(t)}))^{-1}\nabla l(\boldsymbol{\theta}^{(t)})\\ &= \boldsymbol{\theta}^{(t)} + (\sum_{j=1}^{m}\boldsymbol{x}_j\boldsymbol{x}_j^T n_j p_j^{(t)}(1-p_j^{(t)}))^{-1}\sum_{j=1}^{m}\boldsymbol{x}_j(y_j - n_j p_j^{(t)}).\end{aligned}$$

- The Gradient method is

$$\begin{aligned}\boldsymbol{\theta}^{(t+1)} &= \boldsymbol{\theta}^{(t)} + \alpha\nabla l(\boldsymbol{\theta}^{(t)})\\ &= \boldsymbol{\theta}^{(t)} + \alpha\sum_{j=1}^{m}\boldsymbol{x}_j(y_j - n_j p_j^{(t)}),\end{aligned}$$

where α is called the learning rate.

Logistic Regression

Practice - numerical differentiation

- In practice, we may not be able to derive the closed from for the score and the observed information. But you can always derive they numerically!
- A typical example of the numerical derivative of $f(x)$ is

  $$f'(x) \approx \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x},$$

  which is called the central difference approximation. Here the spacing Δx is an important hyper-parameter.
- You can use the python package 'numdifftools'. It conveniently gives us the gradient vector and the Hessian matrix.
- The numerical differentiation can also be used to check that the score or the observed information were derived correctly.
- The problems are that it may be too slow compared to the closed form and that it may sometimes be numerically unstable.

Logistic Regression

Practice - a question

- Draw $\boldsymbol{x}_j \sim N(\mathbf{0}, I_2)$, set $n_j = 10$, $j = 1, \ldots, 20$, and obtain $y_{1:20}$ from the logistic model with $\theta_0 = [-1, 2]^T$. Try to do optimization of the log-likelihood given $\{(y_1, n_1, \boldsymbol{x}_1), \ldots, (y_{20}, n_{20}, \boldsymbol{x}_{20})\}$.
- Compare the convergence speeds of the Newton–Raphson method and the Gradient method. Try different α.

MLE Estimator

MLE Estimator

Problem formulation - the problem

- Remember that, in the Importance Sampling, to estimate the integration $\mu = \mathbb{E}_\pi[f(X)] = \int \pi(\boldsymbol{x}) f(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}$, we propose the weighted samples $\{(\boldsymbol{x}_1, w_1), \ldots, (\boldsymbol{x}_n, w_n)\}$ where $\boldsymbol{x}_i \sim q(\boldsymbol{x})$ and $w_i = \pi(\boldsymbol{x}_i)/q(\boldsymbol{x}_i)$, and construct the estimator $\hat{\mu}_{\mathrm{IS}}(f) = \sum_{i=1}^n w_i f(\boldsymbol{x}_i)/n$.
- Suppose that we know the values of m integration $\int g_j(\boldsymbol{x}) \mathrm{d}x = b_j$, or equivalently $\int \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0}$, where $\boldsymbol{g}(\boldsymbol{x}) = (g_1(\boldsymbol{x}) - b_1 q(\boldsymbol{x}), \ldots, g_m(\boldsymbol{x}) - b_m q(\boldsymbol{x}))^T$.
- The MLE estimator aims to solve the problem that how to use the information $\int \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0}$ to improve the performance of the estimator $\hat{\mu}_{\mathrm{IS}}$.

MLE Estimator

Problem formulation - the definition

- The core of this approach is to solve the nonparametric maximum likelihood problem

$$\hat{\theta} = \underset{\zeta}{\operatorname{argmax}} l(\theta) = \underset{\zeta}{\operatorname{argmax}} \frac{1}{n} \sum_{i=1}^{n} \log(q(\boldsymbol{x}_i) + \theta^T \boldsymbol{g}(\boldsymbol{x}_i)),$$

- The MLE estimator is defined as

$$\hat{\mu}_{\mathrm{MLE}}(f) = \frac{1}{n} \sum_{i=1}^{n} \frac{\pi(\boldsymbol{x}_i)}{q(\boldsymbol{x}_i) + \hat{\theta}^T \boldsymbol{g}(\boldsymbol{x}_i)} f(\boldsymbol{x}_i).$$

- It’s equivalent to do the IS estimation with new weights $w_i' = \pi(\boldsymbol{x}_i)/(q(\boldsymbol{x}_i) + \hat{\theta}^T \boldsymbol{g}(\boldsymbol{x}_i))$.

MLE Estimator

Newton–Raphson method - the score

- Take the gradient of the log-likelihood

$$\begin{aligned}\nabla l(\boldsymbol{\theta}) &= \frac{1}{n}\sum_{i=1}^{n} \nabla \log(q(\boldsymbol{x}_i) + \boldsymbol{\theta}^T \boldsymbol{g}(\boldsymbol{x}_i)) \\ &= \frac{1}{n}\sum_{i=1}^{n} \frac{\boldsymbol{g}(\boldsymbol{x}_i)}{q(\boldsymbol{x}_i) + \boldsymbol{\theta}^T \boldsymbol{g}(\boldsymbol{x}_i)}.\end{aligned}$$

- Notice that at the MLE point, we have $\nabla l(\hat{\boldsymbol{\theta}}) = \mathbf{0}$, which essentially means that $\boldsymbol{g}(\boldsymbol{x})$ has measure **0** with the discrete measure $\hat{\nu}$ defined by

$$\int h(\boldsymbol{x}) \mathrm{d}\hat{\nu} = \frac{1}{n}\sum_{i=1}^{n} \frac{h(\boldsymbol{x}_i)}{q(\boldsymbol{x}_i) + \hat{\boldsymbol{\theta}}^T \boldsymbol{g}(\boldsymbol{x}_i)},$$

for arbitrary $h(\boldsymbol{x})$. And also $\hat{\mu}_{\mathrm{MLE}}(f) = \int \pi(\boldsymbol{x}) f(\boldsymbol{x}) \mathrm{d}\hat{\nu}$.

MLE Estimator

Newton–Raphson method - the observed information

- Take the negative Hessian of the log-likelihood

$$
\begin{aligned}
-\nabla^2 l(\theta) &= -\nabla \frac{1}{n}\sum_{i=1}^{n} \frac{\boldsymbol{g}(\boldsymbol{x}_i)}{q(\boldsymbol{x}_i)+\theta^T \boldsymbol{g}(\boldsymbol{x}_i)} \\
&= -\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{g}(\boldsymbol{x}_i)\nabla \frac{1}{q(\boldsymbol{x}_i)+\theta^T \boldsymbol{g}(\boldsymbol{x}_i)} \\
&= \frac{1}{n}\sum_{i=1}^{n} \frac{\boldsymbol{g}(\boldsymbol{x}_i)\boldsymbol{g}(\boldsymbol{x}_i)^T}{(q(\boldsymbol{x}_i)+\theta^T \boldsymbol{g}(\boldsymbol{x}_i))^2}.
\end{aligned}
$$

MLE Estimator

Newton–Raphson method - the algorithm

- The Newton–Raphson method is

$$\theta^{(t+1)} = \theta^{(t)} - (\nabla^2 l(\theta^{(t)}))^{-1}\nabla l(\theta^{(t)})$$
$$= \theta^{(t)} + (\sum_{i=1}^{n} \frac{\boldsymbol{g}(\boldsymbol{x}_i)\boldsymbol{g}(\boldsymbol{x}_i)^T}{(q(\boldsymbol{x}_i) + \theta^{(t)T}\boldsymbol{g}(\boldsymbol{x}_i))^2})^{-1} \sum_{i=1}^{n} \frac{\boldsymbol{g}(\boldsymbol{x}_i)}{q(\boldsymbol{x}_i) + \theta^{(t)T}\boldsymbol{g}(\boldsymbol{x}_i)}.$$

- If we directly consider the original question

$$\hat{\theta} = \underset{\zeta}{\text{argmax}} l(\theta) = \underset{\zeta}{\text{argmax}} \frac{1}{n}\sum_{i=1}^{n} \log(q(\boldsymbol{x}_i) + \theta^T\boldsymbol{g}(\boldsymbol{x}_i)),$$

  It is actually a constrained optimization problem $(q(\boldsymbol{x}_i) + \theta^T\boldsymbol{g}(\boldsymbol{x}_i) > 0, i = 1, \ldots, n)$.
- It is recommended to use the Sequential Least SQuares Programming (SLSQP) Algorithm to solve the constrained optimization.

MLE Estimator

Practice - a question

- Consider $\pi(\boldsymbol{x})$ to be $N(\mathbf{0}, I_2)$, $q(\boldsymbol{x})$ to be $N(\mathbf{0}, 4I_2)$, and $f(\boldsymbol{x}) = x_1^2 + x_2^2$. Draw 100 samples from $q(\boldsymbol{x})$ and build normal kernels $K_j(\boldsymbol{x})$ with covariance matrix $0.8^2 I_2$ centered at the 100 samples. Define $\boldsymbol{g}(\boldsymbol{x}) = (K_1(\boldsymbol{x}) - q(\boldsymbol{x}), \ldots, K_{100}(\boldsymbol{x}) - q(\boldsymbol{x}))^T$.
- Draw 10000 new samples from $q(\boldsymbol{x})$ and use the MLE estimator to estimate $\mu = \mathbb{E}_\pi[f(X)] = \int \pi(\boldsymbol{x}) f(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}$. Do 100 repetitions to compare the variance of $\hat{\mu}_{\mathrm{MLE}}(f)$ with that of the original Importance Sampling estimator $\hat{\mu}_{\mathrm{IS}}(f)$.

Gaussian Quadrature

Gaussian Quadrature

The basics - the definition

- To approximate the integration $\mu = \mathbb{E}_\pi[f(X)] = \int \pi(x)f(x)\mathrm{d}x$, a methodology called Gaussian Quadrature (GQ) deterministically proposes weighted points $\{(x_1, w_1), \ldots, (x_n, w_n)\}$ so that $\hat{\mu}_{\mathrm{GQ}}(f) = \sum_{i=1}^{n} w_i f(x_i) \approx \mu$.
- The locations of the points are derived by the equation set $\{\mathbb{E}_\pi[p_n(X)X^k] = 0; k = 0, \ldots, n-1\}$, where the $p_n(x) = \prod_{i=1}^{n}(x - x_i)$.
- Given the points, the weights are obtain by another equation set $\{\mathbb{E}_\pi[X^k] = \sum_{i=1}^{n} w_i x_i^k; k = 0, \ldots, n-1\}$.
- The weighted points are uniquely determined by $\pi(x)$.

Gaussian Quadrature

The basics - an intuitive justification

- The GQ estimator $\hat{\mu}_{\mathrm{GQ}}$ has the advantage that for any polynomial $h_{2n-1}(x)$ of order less or equal to $2n-1$, $\hat{\mu}_{\mathrm{GQ}}(h_{2n-1}) = \mathbb{E}_\pi[h_{2n-1}(X)]$ exactly. This is because of that we can decompose $h_{2n-1}(x)$ to be

$$h_{2n-1}(x) = t_{n-1}(x)p_n(x) + r_{n-1}(x).$$

An so,

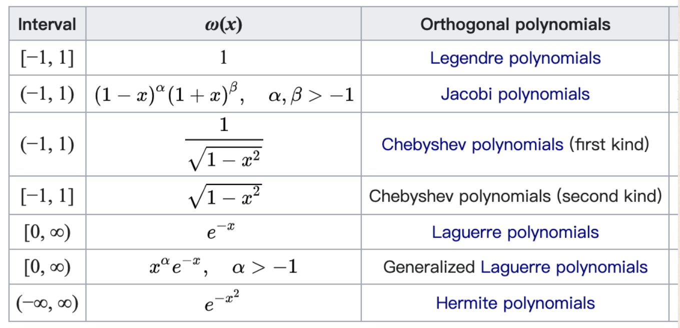
$$\begin{aligned}\hat{\mu}_{\mathrm{GQ}}(h_{2n-1}) &= \sum_{i=1}^{n} w_i h_{2n-1}(x_i) = \sum_{i=1}^{n} w_i(t_{n-1}(x_i)p_n(x_i) + r_{n-1}(x_i)) \\ &= \sum_{i=1}^{n} w_i r_{n-1}(x_i) = \mathbb{E}_\pi[r_{n-1}(X)] \\ &= \mathbb{E}_\pi[t_{n-1}(X)p_n(X)] + \mathbb{E}_\pi[r_{n-1}(X)] = \mathbb{E}_\pi[h_{2n-1}(X)].\end{aligned}$$

- The Stone–Weierstrass theorem states that every continuous function defined on a closed interval can be uniformly approximated by a polynomial function.

Gaussian Quadrature

The basics - three types

- The classic GQ with weight funtion $\omega(x)$ includes



Interval	$\omega(x)$	Orthogonal polynomials
$[-1, 1]$	1	Legendre polynomials
$(-1, 1)$	$(1-x)^{\alpha}(1+x)^{\beta}, \quad \alpha, \beta > -1$	Jacobi polynomials
$(-1, 1)$	$\frac{1}{\sqrt{1-x^2}}$	Chebyshev polynomials (first kind)
$[-1, 1]$	$\sqrt{1-x^2}$	Chebyshev polynomials (second kind)
$[0, \infty)$	e^{-x}	Laguerre polynomials
$[0, \infty)$	$x^{\alpha}e^{-x}, \quad \alpha > -1$	Generalized Laguerre polynomials
$(-\infty, \infty)$	e^{-x^2}	Hermite polynomials

- If we consider $\pi(x) \propto \omega(x)$, there are only three types, the Beta distribution, the Gamma distribution and the Normal distribution.
- They are defined respectively on the finite interval, half infinite interval and infinite interval.

Gaussian Quadrature

Connections with the Importance Sampling - the three phenomena

We can actually view GQ as one kind of deterministic Importance Sampling, which is based on the three phenomena:

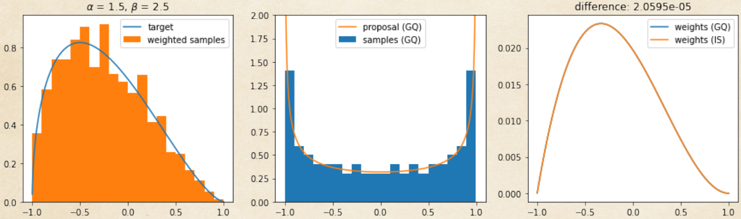
- The weighted empirical cdf of the GQ samples $\hat{\Pi}_n = \sum_{i=1}^n w_i \mathbf{1}(x_i \leq x)$ approximate Π which is the cdf of the target distribution $\pi(x)$ (note that by construction $\sum_{i=1}^n w_i = 1$);
- The unweighted empirical cdf of the GQ samples $\hat{P}_n = \sum_{i=1}^n \mathbf{1}(x_i \leq x)/n$ approximate a fixed distribution P (assume $p(x)$ to be the corresponding pdf) determined by $\pi(x)$, and we term $p(x)$ as the GQ proposal;
- The GQ weights have $w_i \approx w_i^{\mathrm{IS}}$, where w_i^{IS} is the normalized weight at point x_i for Importance Sampling, that is $w_i^{\mathrm{IS}} = \frac{\pi(x_i)}{p(x_i)} / \sum_{i=1}^n \frac{\pi(x_i)}{p(x_i)}$.

These phenomena are not trivial. Remember the MCMC points.

Gaussian Quadrature

Connections with the Importance Sampling - the Beta case

- For the Beta distribution $\pi(x) \propto (1+x)^{\alpha-1}(1-x)^{\beta-1}$ defined on the interval $[-1,1]$, P is invariant against α, β. To be more specific, $p(x) \propto 1/\sqrt{1-x^2}$, the Arcsine distribution.

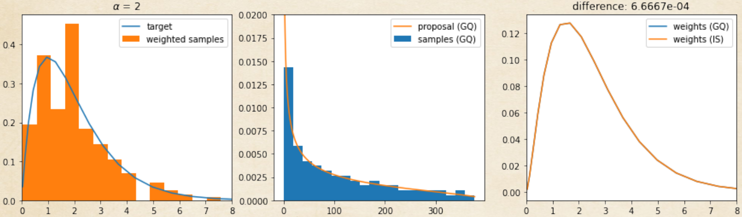


- Actually, this GQ proposal is true for all $\pi(x)$ that is positive and satisfies some regularity conditions, a.e., on a compact interval.

Gaussian Quadrature

Connections with the Importance Sampling - the Gamma case

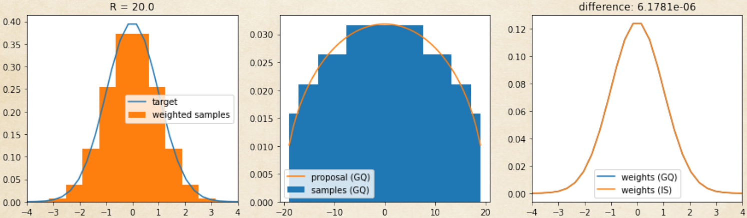
- Consider the Gamma distribution with pdf $\pi(x) \propto x^{\alpha-1}e^{-x}$. There would be $p(x) \propto \sqrt{(4n-x)/x}$, which has an expanding support $[0, 4n]$, and this GQ proposal is also invariant against α. It is called the Marchenko–Pastur distribution.



Gaussian Quadrature

Connections with the Importance Sampling - the Normal case

- If $\pi(x) \propto e^{-x^2/2}$ (the Normal case), the support of P also expands as n increases, but keeps the same shape: $p(x) \propto \sqrt{R_n^2 - x^2}$, $R_n = 2\sqrt{n}$, the Wigner Semicircle distribution.



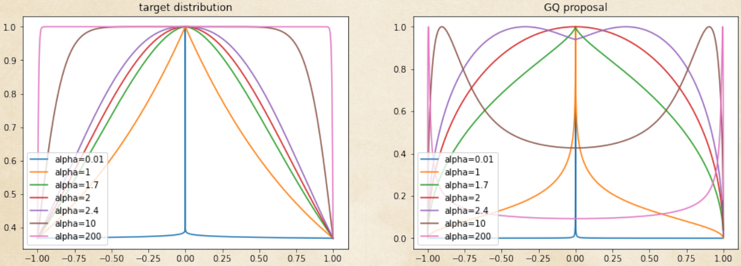
- Given the finite interval case, you may wonder if the P is also invariant against different $\pi(\boldsymbol{x})$ on the half infinite or infinite intervals, but this is actually not true.

Gaussian Quadrature

Connections with the Importance Sampling - the Ullman distribution

- The Ullman distribution is a strange distribution, you can't even find it on Wikipedia!
- Given the target distribution $\pi(x) \propto e^{-|x|^{\alpha}}$, the support of its GQ proposal expands at the speed of $n^{1/\alpha}$ and has the shape of the Ullman distribution

$$p(x) = \frac{\alpha}{\pi} \int_{|x|}^{1} \frac{y^{\alpha-1}}{\sqrt{y^2 - x^2}} \mathrm{d}y.$$



Gaussian Quadrature

Practice - a question

- Try to find out the counterpart of the Ullman distribution on the half infinite interval, where the Marchenko–Pastur distribution should be its special case and the Arcsine distribution should be its limit.

Quasi Monte Carlo

Quasi Monte Carlo

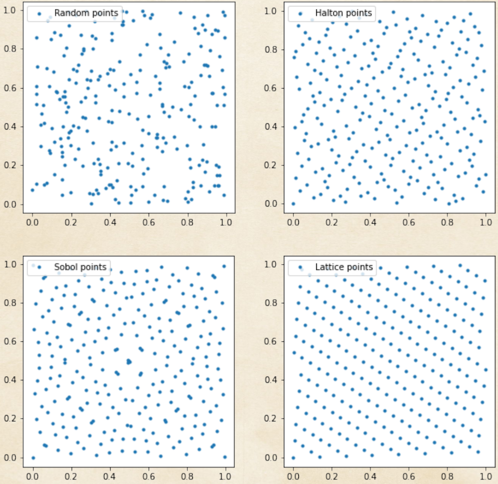
Problem formulation - the curse of dimensionality

- In the numerical integration, the deterministic methods reach an accuracy of order $n^{-k/d}$, where k stands for the regularity of the integrand and d is the dimension of the integration domain, whereas the classic Monte Carlo is subjected to an error bound of order $n^{-1/2}$.
- The Quasi Monte Carlo (QMC) aims to achieve n^{-1} for integration on the hypercube $[0, 1]^d$. It is sometimes accurate even when $d > 100$.

Quasi Monte Carlo

Problem formulation - a visualization

- The QMC points are just more uniformly distributed than the random points.



Quasi Monte Carlo

Theories - the discrepancy measure

- To establish the LLN and CLT for the QMC, people use the discrepancy theory that measures the uniformness of points.
- The local discrepancy of $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ at $\boldsymbol{a} \in [0,1]^d$ is

$$\delta(\boldsymbol{a}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(\boldsymbol{x}_i \in [\mathbf{0}, \boldsymbol{a})) - \mathrm{vol}([\mathbf{0}, \boldsymbol{a})).$$

- The star discrepancy of $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ is

$$D^*(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \sup_{\boldsymbol{a}\in[0,1]^d} |\delta(\boldsymbol{a}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)|.$$

Quasi Monte Carlo

Theories - the LLN

- For the infinite sequence $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots\}$, if $\lim_{n\to\infty} D^*(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = 0$, then for Riemann integrable function $f(\boldsymbol{x})$, we have the LLN

$$\left| \frac{1}{n} \sum_{i=1}^{n} f(\boldsymbol{x}_i) - \int_{[0,1]^d} f(\boldsymbol{x}) \mathrm{d}x \right| \to 0.$$

- QMC points can typically achieve $D^*(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \mathcal{O}(\log(n)^d / n)$ and thus satisfy the LLN.

Quasi Monte Carlo

Theories - the CLT

- The Koksma-Hlawka inequality (CLT for QMC) states that for the points $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$,

$$\left| \frac{1}{n} \sum_{i=1}^{n} f(\boldsymbol{x}_i) - \int_{[0,1]^d} f(\boldsymbol{x}) \mathrm{d}x \right| \leq D^*(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) V_{\mathrm{HK}}(f),$$

  where $V_{\mathrm{KH}}(f)$ denotes the total variation of $f(\boldsymbol{x})$ in the sense of Hardy and Krause.
- Remember that $D^*(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \mathcal{O}(\log(n)^d/n)$, so the QMC can converge at the speed $n^{\varepsilon-1}$ for any small $\varepsilon > 0$.
- If you compare it with the classic CLT

$$\frac{1}{n} \sum_{i=1}^{n} f(\boldsymbol{x}_i) - \int_{[0,1]^d} f(\boldsymbol{x}) \mathrm{d}x = n^{-1/2} N(0, \sigma^2(f)),$$

  where $\sigma^2(f) = \mathrm{var}(f(\boldsymbol{X}))$, $\boldsymbol{X} \sim \mathcal{U}([0,1]^d)$.

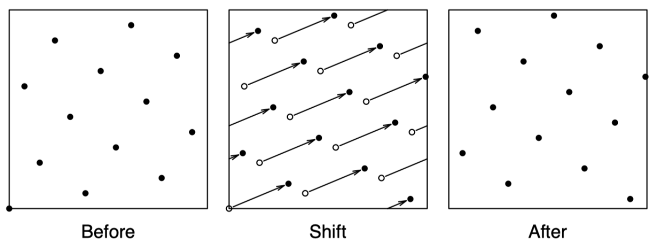
Quasi Monte Carlo

Practice - the python package

- Install and learn how to use the python package ‘qmcpy’.
- In the package, the QMC points can be randomized based on methods like the Cranley-Patterson rotation

$$\boldsymbol{x}_i' = \boldsymbol{x}_i + \boldsymbol{U} \mod 1, \;\; i = 1, \dots, n$$

where $\boldsymbol{U} \sim \mathcal{U}([0,1]^d)$. The Randomized QMC is more robust can sometimes achieve even faster convergence rate.



References

- Tan, Z. (2004). On a likelihood approach for Monte Carlo integration. Journal of the American Statistical Association, 99(468), 1027-1036.
- Dette, H., & Studden, W. J. (1992). On a new characterization of the classical orthogonal polynomials. Journal of approximation theory, 71(1), 3-17.
- Szeg, G. (1939). Orthogonal polynomials (Vol. 23). American Mathematical Soc.
- Freud, G. (2014). Orthogonal polynomials. Elsevier.
- Owen, A. B. (2013). Monte Carlo theory, methods and examples.
- Novak, E. (2016). Some results on the complexity of numerical integration. In Monte Carlo and Quasi-Monte Carlo Methods (pp. 161-183). Springer, Cham.

Thanks!