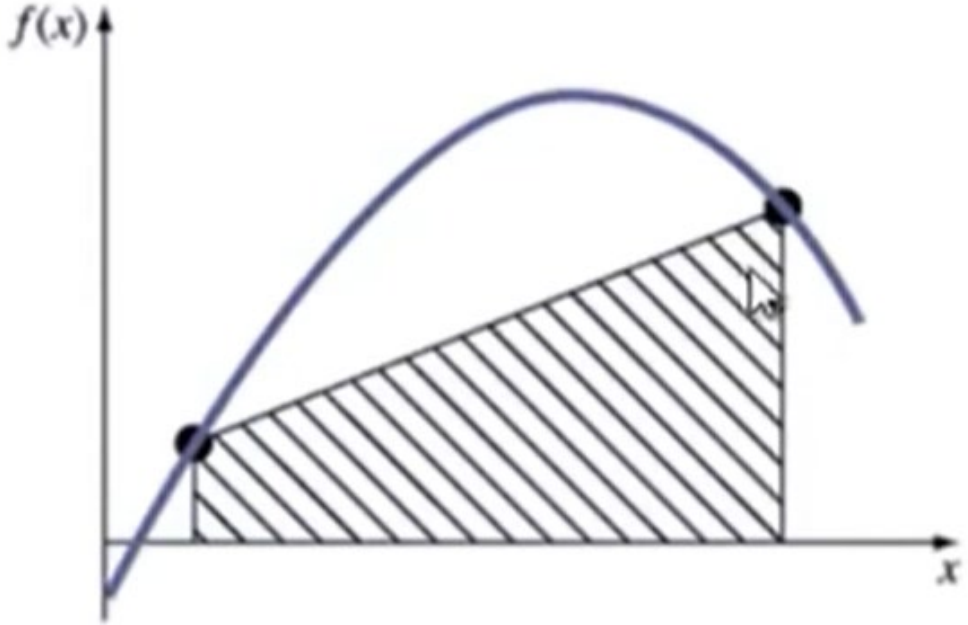
Applied Probability and Statistics

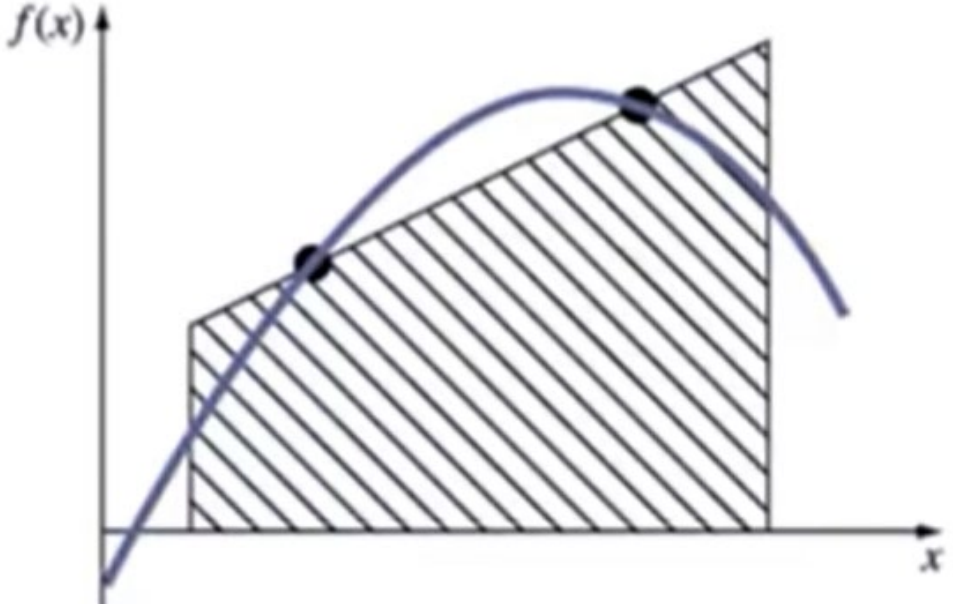
Module-4, Lecture-1

By: Dr. A K Yadav (9911375598)
Dept of CSE, ASET, AUUP

- We will discuss:
- What is Gaussian quadrature
- The conceptual basis for Gauss quadrature
- How Gauss quadrature constants can be determined
- Use Gauss quadrature to calculate the integral of a function

- Evaluate the area under a straight line by joining any two points on a curve rather than simply choosing the endpoints.
- The key is to choose the line that balances the positive and negative errors.





Gaussian Quadrature

- The Gaussian quadrature method is an approximate method of calculation of a certain integral $I = \int_a^b f(x)dx$
- By replacing the variables $x = \frac{(b-a)}{2}t + \frac{(b+a)}{2}$, $dx = \frac{(b-a)}{2}dt$ the desired integral is reduced to the form $I = \int_{-1}^{1} f(\frac{(b-a)}{2}t + \frac{(b+a)}{2})\frac{(b-a)}{2}dt$

- The Gaussian quadrature formula is $\int_{-1}^{1} f(x)dx = \sum_{i=1}^{n} A_i f(x_i)$
- The nodes x_i of the Gaussian quadrature formula are the roots of a Legendre polynomial $P_n(x)$ of degree n.
- The Legendre polynomial has exactly n real different roots in the interval $(-1,1)$.
- The weights A_i of the Gaussian quadrature formula are defined by
- $A_i = \frac{2}{(1-x_i^2)[P_n'(x_i)]^2}$

# Points	Weighting Factors, c_i	Function Arguments, x_i
2	$c_0 = 1.0$ $c_1 = 1.0$	$x_0 = -1/\sqrt{3}$ $x_1 = 1/\sqrt{3}$
3	$c_0 = 5/9$ $c_1 = 8/9$ $c_2 = 5/9$	$x_0 = -\sqrt{3/5}$ $x_1 = 0.0$ $x_2 = \sqrt{3/5}$
4	$c_0 = (18-\sqrt{30})/36$ $c_1 = (18+\sqrt{30})/36$ $c_2 = (18+\sqrt{30})/36$ $c_3 = (18-\sqrt{30})/36$	$x_0 = -\sqrt{525+70\sqrt{30}}/35$ $x_1 = -\sqrt{525-70\sqrt{30}}/35$ $x_2 = \sqrt{525-70\sqrt{30}}/35$ $x_3 = \sqrt{525+70\sqrt{30}}/35$

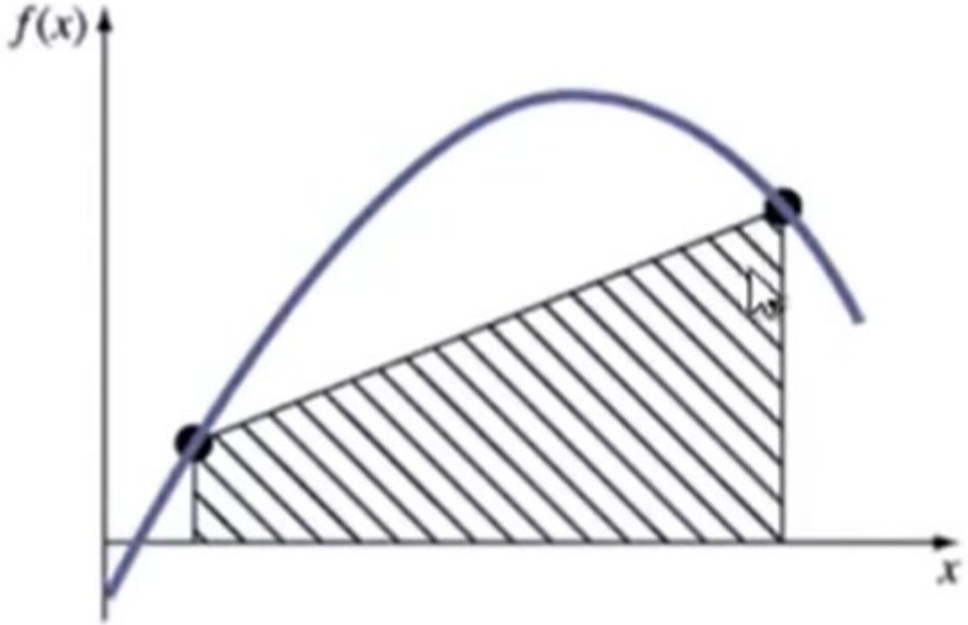
Applied Probability and Statistics

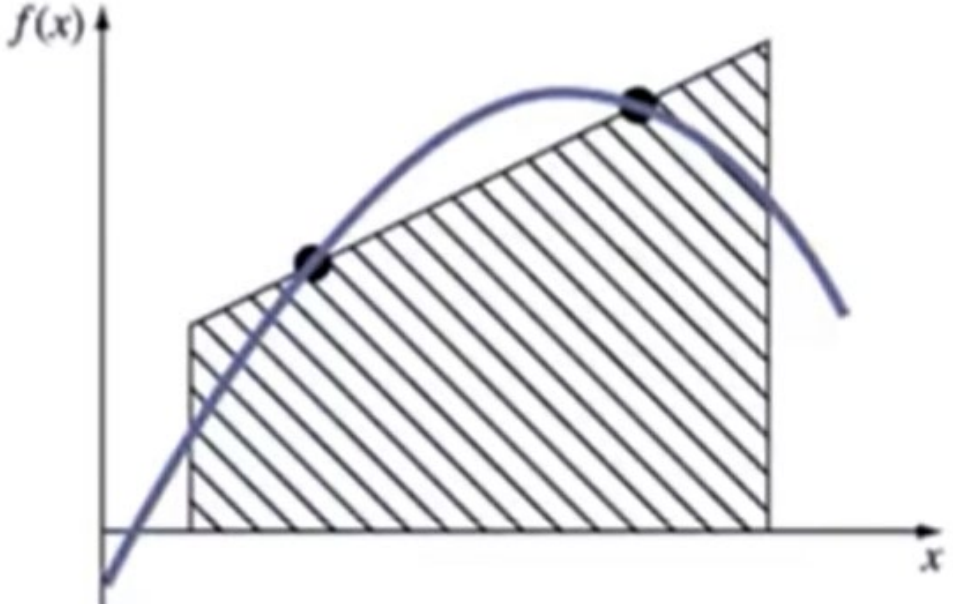
Module-4, Lecture-2

By: Dr. A K Yadav (9911375598)
Dept of CSE, ASET, AUUP

- We will discuss:
- What is Gaussian quadrature
- The conceptual basis for Gauss quadrature
- How Gauss quadrature constants can be determined
- Use Gauss quadrature to calculate the integral of a function

- Evaluate the area under a straight line by joining any two points on a curve rather than simply choosing the endpoints.
- The key is to choose the line that balances the positive and negative errors.





Gaussian Quadrature

- The Gaussian quadrature method is an approximate method of calculation of a certain integral $I = \int_a^b f(x)dx$
- By replacing the variables $x = \frac{(b-a)}{2}t + \frac{(b+a)}{2}$, $dx = \frac{(b-a)}{2}dt$ the desired integral is reduced to the form $I = \int_{-1}^{1} f(\frac{(b-a)}{2}t + \frac{(b+a)}{2})\frac{(b-a)}{2}dt$

- The Gaussian quadrature formula is $\int_{-1}^{1} f(x)dx = \sum_{i=1}^{n} A_i f(x_i)$
- The nodes x_i of the Gaussian quadrature formula are the roots of a Legendre polynomial $P_n(x)$ of degree n.
- The Legendre polynomial has exactly n real different roots in the interval $(-1,1)$.
- The weights A_i of the Gaussian quadrature formula are defined by
- $A_i = \frac{2}{(1-x_i^2)[P_n'(x_i)]^2}$

# Points	Weighting Factors, c_i	Function Arguments, x_i
2	$c_0 = 1.0$ $c_1 = 1.0$	$x_0 = -1/\sqrt{3}$ $x_1 = 1/\sqrt{3}$
3	$c_0 = 5/9$ $c_1 = 8/9$ $c_2 = 5/9$	$x_0 = -\sqrt{3/5}$ $x_1 = 0.0$ $x_2 = \sqrt{3/5}$
4	$c_0 = (18-\sqrt{30})/36$ $c_1 = (18+\sqrt{30})/36$ $c_2 = (18+\sqrt{30})/36$ $c_3 = (18-\sqrt{30})/36$	$x_0 = -\sqrt{525+70\sqrt{30}}/35$ $x_1 = -\sqrt{525-70\sqrt{30}}/35$ $x_2 = \sqrt{525-70\sqrt{30}}/35$ $x_3 = \sqrt{525+70\sqrt{30}}/35$

Department of Computer Science and Engineering

Applied Probability and Statistics

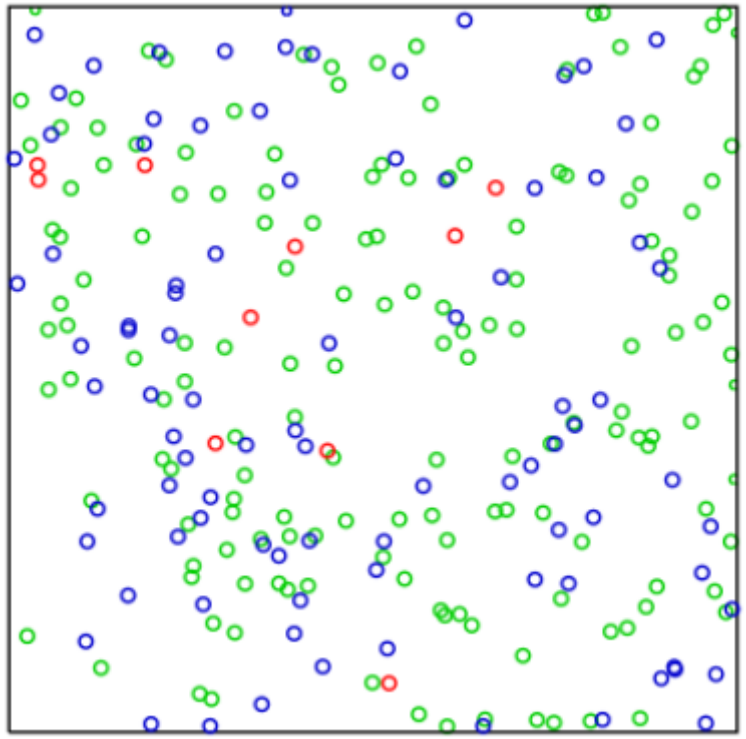
Module-4, Lecture-3

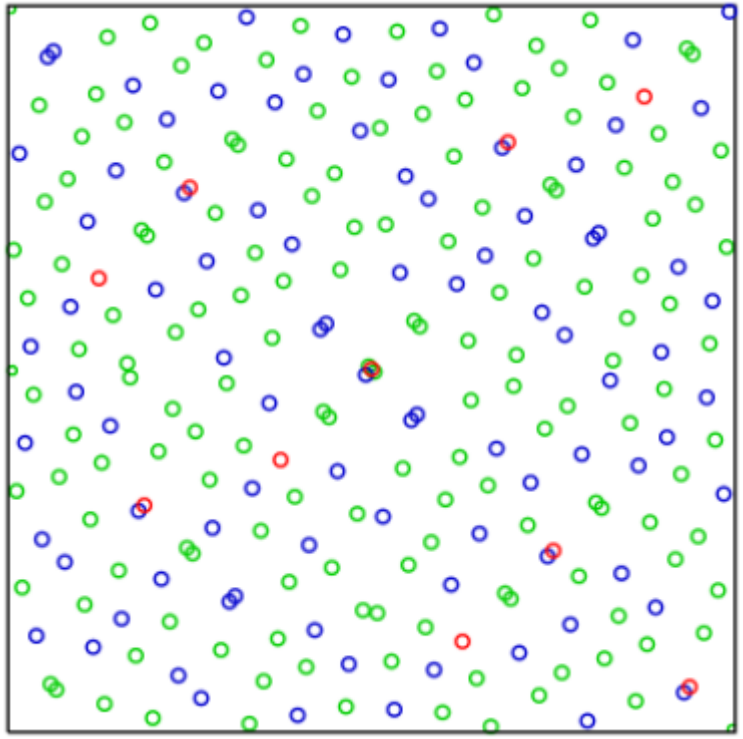
By: Dr. A K Yadav (9911375598)
Dept of CSE, ASET, AUUP

quasi-Monte Carlo

- The quasi-Monte Carlo method is a method for numerical integration and solving problems using low-discrepancy sequences ,also called quasi-random sequences or sub-random sequences.
- This contrasts with the regular Monte Carlo method or Monte Carlo integration, which are based on sequences of pseudorandom numbers.
- Monte Carlo and quasi-Monte Carlo methods are stated in a similar way.

- Pseudorandom sequence (Fig: Left)
- A Sobol sequence of low-discrepancy quasi-random numbers





- The problem is to approximate the integral of a function f as the average of the function evaluated at a set of points $x_1, x_2, \ldots, x_N$
- $\int_{[0,1]^s} f(u)du \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)$
- We are integrating over the s-dimensional unit cube, so each x_i is a vector of s elements.
- The difference between quasi-Monte Carlo and Monte Carlo is the way the x_i is chosen.
- Quasi-Monte Carlo uses a low-discrepancy sequence such as the Halton sequence, the Sobol sequence, or the Faure sequence, whereas Monte Carlo uses a pseudorandom sequence.

- The advantage of using low-discrepancy sequences is a faster rate of convergence.
- Quasi-Monte Carlo has a rate of convergence close to $O(\frac{1}{N})$, whereas the rate for the Monte Carlo method is $O(\frac{1}{\sqrt{N}})$
- The Quasi-Monte Carlo method recently became popular in the area of mathematical finance or computational finance.
- In these areas, high-dimensional numerical integrals, where the integral should be evaluated within a threshold ε, occur frequently.
- Hence, the Monte Carlo method and the quasi-Monte Carlo method are beneficial in these situations.

Approximation error

- The approximation error of the quasi-Monte Carlo method is bounded by a term proportional to the discrepancy of the set $x_1, \ldots . x_N$.
- Specifically, the Koksma–Hlawka inequality states that the error
- $\varepsilon = \left| \int_{[0,1]^s} f(u)du \approx \frac{1}{N} \sum_{i=1}^{N} f(x_i) \right|$ is bounded by $|\varepsilon| \leq V(f) D_N^*$
- Where $V(f)$ is the Hardy–Krause variation of the function f and D_N^* is the called star discrepancy of the set $x_1, \ldots, x_N$ and is defined as
- $D_N^* = \sup_{Q \subset [0,1]^s} \left| \frac{No\ of\ points\ in\ Q}{N} - Volume(Q) \right|$

- Where Q is a rectangular solid in $[0,1]^s$ with sides parallel to the coordinate axes.
- The inequality $|\varepsilon| \leq V(f)D_N^*$ can be used to show that the error of the approximation by the quasi-Monte Carlo method is $O(\frac{(\log N)^S}{N})$, whereas the Monte Carlo method has a probabilistic error of $O(\frac{1}{\sqrt{N}})$
- We can only state the upper bound of the approximation error, the convergence rate of quasi-Monte Carlo method in practice is usually much faster than its theoretical bound.

- In general, the accuracy of the quasi-Monte Carlo method increases faster than that of the Monte Carlo method.
- However, this advantage is only guaranteed if N is large enough and if the variation is finite.

Drawbacks of quasi-Monte Carlo

- In order for $O(\frac{(\log N)^S}{N})$ to be smaller than $O(\frac{1}{\sqrt{N}})$, s needs to be small and N needs to be large e.g., $N > 2^s$. For large s and practical N values, the discrepancy of a point set from a low-discrepancy generator might be not smaller than for a random set.
- For many functions arising in practice, $V(f) = \infty$, e.g., if Gaussian variables are used.
- We only know an upper bound on the error and it is difficult to compute D_N^* and $V(f)$.

Applied Probability and Statistics

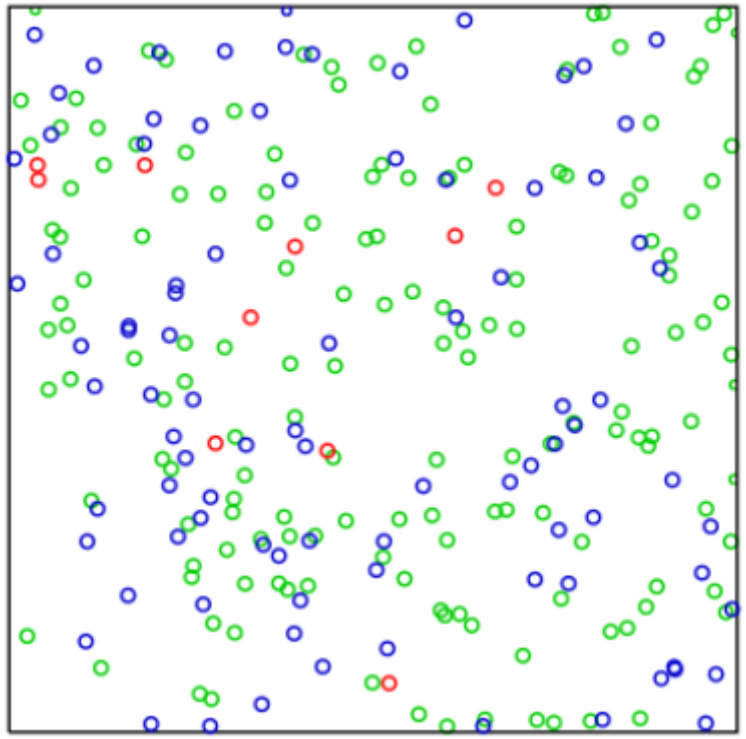
Module-4, Lecture-4

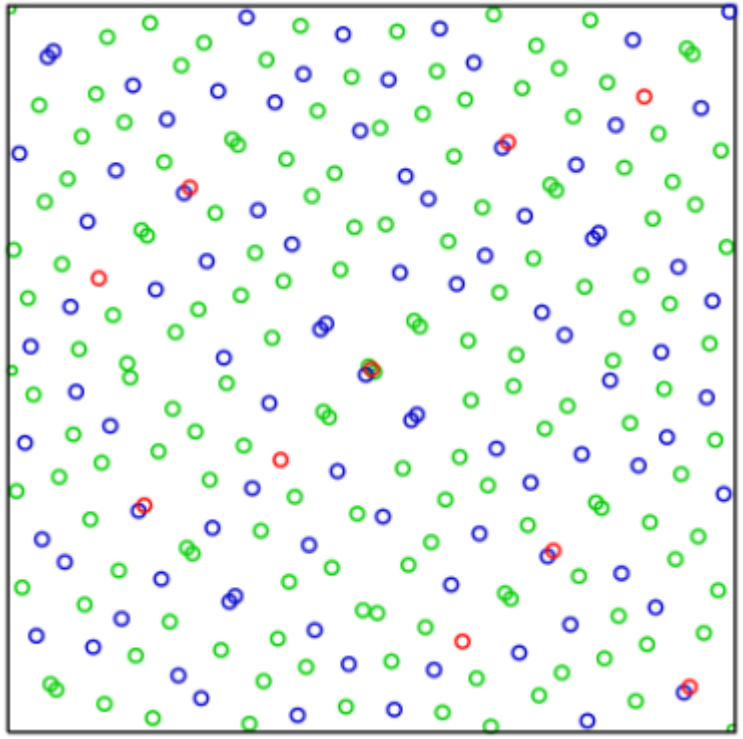
By: Dr. A K Yadav (9911375598)
Dept of CSE, ASET, AUUP

quasi-Monte Carlo

- The quasi-Monte Carlo method is a method for numerical integration and solving problems using low-discrepancy sequences ,also called quasi-random sequences or sub-random sequences.
- This contrasts with the regular Monte Carlo method or Monte Carlo integration, which are based on sequences of pseudorandom numbers.
- Monte Carlo and quasi-Monte Carlo methods are stated in a similar way.

- Pseudorandom sequence (Fig: Left)
- A Sobol sequence of low-discrepancy quasi-random numbers





- The problem is to approximate the integral of a function f as the average of the function evaluated at a set of points $x_1, x_2, \ldots, x_N$
- $\int_{[0,1]^s} f(u)du \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)$
- We are integrating over the s-dimensional unit cube, so each x_i is a vector of s elements.
- The difference between quasi-Monte Carlo and Monte Carlo is the way the x_i is chosen.
- Quasi-Monte Carlo uses a low-discrepancy sequence such as the Halton sequence, the Sobol sequence, or the Faure sequence, whereas Monte Carlo uses a pseudorandom sequence.

- The advantage of using low-discrepancy sequences is a faster rate of convergence.
- Quasi-Monte Carlo has a rate of convergence close to $O(\frac{1}{N})$, whereas the rate for the Monte Carlo method is $O(\frac{1}{\sqrt{N}})$
- The Quasi-Monte Carlo method recently became popular in the area of mathematical finance or computational finance.
- In these areas, high-dimensional numerical integrals, where the integral should be evaluated within a threshold ε, occur frequently.
- Hence, the Monte Carlo method and the quasi-Monte Carlo method are beneficial in these situations.

Approximation error

- The approximation error of the quasi-Monte Carlo method is bounded by a term proportional to the discrepancy of the set $x_1, \ldots . x_N$.
- Specifically, the Koksma–Hlawka inequality states that the error
- $\varepsilon = |\int_{[0,1]^s} f(u)du - \frac{1}{N}\sum_{i=1}^{N} f(x_i)|$ is bounded by $|\varepsilon| \leq V(f)D_N^*$
- Where $V(f)$ is the Hardy–Krause variation of the function f and D_N^* is the called star discrepancy of the set $x_1, \ldots, x_N$ and is defined as
- $D_N^* = \sup_{Q \subset [0,1]^s} |\frac{No\ of\ points\ in\ Q}{N} - Volume(Q)|$

- Where Q is a rectangular solid in $[0,1]^s$ with sides parallel to the coordinate axes.
- The inequality $|\varepsilon| \leq V(f)D_N^*$ can be used to show that the error of the approximation by the quasi-Monte Carlo method is $O(\frac{(\log N)^S}{N})$, whereas the Monte Carlo method has a probabilistic error of $O(\frac{1}{\sqrt{N}})$
- We can only state the upper bound of the approximation error, the convergence rate of quasi-Monte Carlo method in practice is usually much faster than its theoretical bound.

- In general, the accuracy of the quasi-Monte Carlo method increases faster than that of the Monte Carlo method.
- However, this advantage is only guaranteed if N is large enough and if the variation is finite.

Drawbacks of quasi-Monte Carlo

- In order for $O(\frac{(\log N)^S}{N})$ to be smaller than $O(\frac{1}{\sqrt{N}})$, s needs to be small and N needs to be large e.g., $N > 2^s$. For large s and practical N values, the discrepancy of a point set from a low-discrepancy generator might be not smaller than for a random set.
- For many functions arising in practice, $V(f) = \infty$, e.g., if Gaussian variables are used.
- We only know an upper bound on the error and it is difficult to compute D_N^* and $V(f)$.

Applied Probability and Statistics

Module-4, Lecture-5

By: Dr. A K Yadav (9911375598)
Dept of CSE, ASET, AUUP

Expectation propagation

- Expectation Propagation (EP) unifies two previous techniques: assumed-density filtering (ADF), an extension of the Kalman filter, and loopy belief propagation (LBP), an extension of belief propagation in Bayesian networks.
- Loopy belief propagation, because it propagates exact belief states, is useful for a limited class of belief networks, such as those which are purely discrete.
- Expectation Propagation approximates the belief states by only retaining expectations, such as mean and variance, and iterates until these expectations are consistent throughout the network.

- This makes it applicable to hybrid networks with discrete and continuous nodes.
- Experiments with Gaussian mixture models show Expectation Propagation to be convincingly better than methods with similar computational cost: Laplace's method, variational Bayes, and Monte Carlo.
- Expectation Propagation also provides an efficient algorithm for training Bayes point machine classifiers.

ADF algorithm

1. Initialize $\mathbf{m}_x = \mathbf{0}$, $v_x = 100$ (the prior). Initialize $s = 1$ (the scale factor).

2. For each data point $\mathbf{y}_i$, update $(\mathbf{m}_x, v_x, s)$ according to

$$
\begin{aligned}
s &= s^{\backslash i} \times Z_i \\
r_i &= 1 - \frac{1}{Z_i} w \mathcal{N}(\mathbf{y}_i; \mathbf{0}, 10\mathbf{I}) \\
\mathbf{m}_x &= \mathbf{m}_x^{\backslash i} + v_x^{\backslash i} r_i \frac{\mathbf{y}_i - \mathbf{m}_x^{\backslash i}}{v_x^{\backslash i} + 1} \\
v_x &= v_x^{\backslash i} - r_i \frac{(v_x^{\backslash i})^2}{v_x^{\backslash i} + 1} + r_i(1 - r_i) \frac{(v_x^{\backslash i})^2 \|\mathbf{y}_i - \mathbf{m}_x^{\backslash i}\|^2}{d(v_x^{\backslash i} + 1)^2}
\end{aligned}
$$

1. Initialize the term approximations $\tilde{t}_i$
2. Compute the posterior for $\mathbf{x}$ from the product of $\tilde{t}_i$:

$$q(\mathbf{x}) = \frac{\prod_i \tilde{t}_i(\mathbf{x})}{\int \prod_i \tilde{t}_i(\mathbf{x}) d\mathbf{x}} \tag{9}$$

3. Until all $\tilde{t}_i$ converge:
   (a) Choose a $\tilde{t}_i$ to refine
   (b) Remove $\tilde{t}_i$ from the posterior to get an 'old' posterior $q^{\setminus i}(\mathbf{x})$, by dividing and normalizing:

$$q^{\setminus i}(\mathbf{x}) \propto \frac{q(\mathbf{x})}{\tilde{t}_i(\mathbf{x})} \tag{10}$$

(c) Combine $q^{\setminus i}(\mathbf{x})$ and $t_i(\mathbf{x})$ and minimize KL-divergence to get a new posterior $q(\mathbf{x})$ with normalizer Z_i.

(d) Update $\tilde{t}_i = Z_i q(\mathbf{x})/q^{\setminus i}(\mathbf{x})$.

4. Use the normalizing constant of $q(\mathbf{x})$ as an approximation to $p(D)$:

$$p(D) \approx \int \prod_i \tilde{t}_i(\mathbf{x}) d\mathbf{x} \tag{11}$$

Applied Probability and Statistics

Module-4, Lecture-6

By: Dr. A K Yadav (9911375598)
Dept of CSE, ASET, AUUP

Expectation propagation

- Expectation Propagation (EP) unifies two previous techniques: assumed-density filtering (ADF), an extension of the Kalman filter, and loopy belief propagation (LBP), an extension of belief propagation in Bayesian networks.
- Loopy belief propagation, because it propagates exact belief states, is useful for a limited class of belief networks, such as those which are purely discrete.
- Expectation Propagation approximates the belief states by only retaining expectations, such as mean and variance, and iterates until these expectations are consistent throughout the network.

- This makes it applicable to hybrid networks with discrete and continuous nodes.
- Experiments with Gaussian mixture models show Expectation Propagation to be convincingly better than methods with similar computational cost: Laplace's method, variational Bayes, and Monte Carlo.
- Expectation Propagation also provides an efficient algorithm for training Bayes point machine classifiers.

ADF algorithm

1. Initialize $\mathbf{m}_x = \mathbf{0}$, $v_x = 100$ (the prior). Initialize $s = 1$ (the scale factor).

2. For each data point $\mathbf{y}_i$, update $(\mathbf{m}_x, v_x, s)$ according to

$$
\begin{aligned}
s &= s^{\backslash i} \times Z_i \\
r_i &= 1 - \frac{1}{Z_i} w \mathcal{N}(\mathbf{y}_i; \mathbf{0}, 10\mathbf{I}) \\
\mathbf{m}_x &= \mathbf{m}_x^{\backslash i} + v_x^{\backslash i} r_i \frac{\mathbf{y}_i - \mathbf{m}_x^{\backslash i}}{v_x^{\backslash i} + 1} \\
v_x &= v_x^{\backslash i} - r_i \frac{(v_x^{\backslash i})^2}{v_x^{\backslash i} + 1} + r_i(1 - r_i) \frac{(v_x^{\backslash i})^2 \|\mathbf{y}_i - \mathbf{m}_x^{\backslash i}\|^2}{d(v_x^{\backslash i} + 1)^2}
\end{aligned}
$$

1. Initialize the term approximations $\tilde{t}_i$
2. Compute the posterior for $\mathbf{x}$ from the product of $\tilde{t}_i$:

$$q(\mathbf{x}) = \frac{\prod_i \tilde{t}_i(\mathbf{x})}{\int \prod_i \tilde{t}_i(\mathbf{x}) d\mathbf{x}} \tag{9}$$

3. Until all $\tilde{t}_i$ converge:
   (a) Choose a $\tilde{t}_i$ to refine
   (b) Remove $\tilde{t}_i$ from the posterior to get an 'old' posterior $q^{\setminus i}(\mathbf{x})$, by dividing and normalizing:

$$q^{\setminus i}(\mathbf{x}) \propto \frac{q(\mathbf{x})}{\tilde{t}_i(\mathbf{x})} \tag{10}$$

(c) Combine $q^{\setminus i}(\mathbf{x})$ and $t_i(\mathbf{x})$ and minimize KL-divergence to get a new posterior $q(\mathbf{x})$ with normalizer Z_i.

(d) Update $\tilde{t}_i = Z_i q(\mathbf{x})/q^{\setminus i}(\mathbf{x})$.

4. Use the normalizing constant of $q(\mathbf{x})$ as an approximation to $p(D)$:

$$p(D) \approx \int \prod_i \tilde{t}_i(\mathbf{x}) d\mathbf{x} \tag{11}$$