NORMALIZED IMPORTANCE SAMPLING and LATIN HYPERCUBE SAMPLING

Normalized Importance Sampling :

- Simple importance sampling: we use

$$\theta = E[h(\mathbf{X})] = \int f(\mathbf{x})h(\mathbf{x})d\mathbf{x}$$

$$= E_g[\frac{f(\mathbf{X})h(\mathbf{X})}{g(\mathbf{X})}] = \int g(\mathbf{x})\frac{f(\mathbf{x})h(\mathbf{x})}{g(\mathbf{x})}d\mathbf{x} \approx \theta_{im} = \frac{1}{N}\sum_{i=1}^{N}\frac{f(\mathbf{X}_i)h(\mathbf{X}_i)}{g(\mathbf{X}_i)},$$

with $\mathbf{X}_i \sim G(\mathbf{X})$, for the associated cdf $G(\mathbf{X})$ for pdf $g(\mathbf{x})$.

- Normalized importance sampling: use

$$\theta \approx \theta_{nim} = \frac{1}{N}\sum_{i=1}^{N}\frac{f(\mathbf{X}_i)h(\mathbf{X}_i)}{g(\mathbf{X}_i)} \Big/ \frac{1}{N}\sum_{i=1}^{N}\frac{f(\mathbf{X}_i)}{g(\mathbf{X}_i)},$$

also with $\mathbf{X}_i \sim G(\mathbf{X})$, for associated cdf $G(\mathbf{X})$ for pdf $g(\mathbf{x})$.

Notes:
a) if $f(\mathbf{x})$ is a pdf $\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}\frac{f(\mathbf{X}_i)}{g(\mathbf{X}_i)} = 1$, $\lim_{N\to\infty}\theta_{nim} = \theta$.
b) Normalized importance sampling can be used when f is not normalized.

NORMALIZED IMPORTANCE SAMPLING CONT.

- NIM Example Expected Value Problem:

$$\theta = E[x] = K \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x \ e^{-(x^2y^2+x^2+y^2-8x-8y)/2} dxdy,$$

with $K^{-1} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-(x^2y^2+x^2+y^2-8x-8y)/2} dxdy,$

Raw NIM MC uses $X_{ij} \sim Normal(4,1)$, to estimate

$$\theta \approx \sum_{j=1}^{N} X_{1j} e^{-X_{1j}^2 X_{2j}^2/2} \Big/ \sum_{j=1}^{N} e^{-X_{1j}^2 X_{2j}^2/2}.$$

Matlab:

```
N = 100000; X = 4 + randn(2,N);
F = exp(-prod(X).^2/2); E = X(1,:).*F/mean(F);
disp([mean(E) 2*std(E)/sqrt(N)])
      1.9926      0.64637
N = 1000000; X = 4 + randn(2,N);
disp([mean(E) 2*std(E)/sqrt(N)])
      1.9634      0.20693
```

NORMALIZED IMPORTANCE SAMPLING CONT.

- NIM Example: if $X_i \sim Exp(\frac{1}{i+2})$, $i = 1, 2, 3, 4$, $S(\mathbf{X}) = \sum_{i=1}^4 X_i$, estimate

$$\theta = E[S] = K \int_0^\infty \cdots \int_0^\infty \frac{e^{-\sum_{i=1}^4 \frac{x_i}{i+2}}}{3 \cdot 4 \cdot 5 \cdot 6} S(\mathbf{x}) I\{S(\mathbf{x}) > 62\}\, d\mathbf{x},$$

with $K^{-1} = \int_0^\infty \cdots \int_0^\infty \frac{e^{-\sum_{i=1}^4 \frac{x_i}{i+2}}}{3 \cdot 4 \cdot 5 \cdot 6} I\{S(\mathbf{x}) > 62\}\, d\mathbf{x}$.

Raw NIM MC uses $X_{ij} \sim Exp(\frac{1}{i+2})$, to estimate $\theta \approx \frac{\sum_{j=1}^N S(\mathbf{X}_j) I\{S(\mathbf{X}_j) > 62\}}{\sum_{j=1}^N I\{S(\mathbf{X}_j) > 62\}}$.

Matlab:

```
N = 100000; U = rand(4,N); X = -diag([3:6])*log(1-U);
S = sum(X); h = S.*( S > 62 ); E = h/mean( ( S > 62 ) );
disp([mean(E) 2*std(E)/sqrt(N)])
        66.948        15.237
```

Note: large error, and error estimate also includes errors in K.

NORMALIZED IMPORTANCE SAMPLING CONT.

NIM Example continued:

try tilted density, using common tilt parameter t, so that $X_i \sim Exp(1/(i+2)-t)$,

$$\theta = \prod_{i=1}^{4} \frac{i+2}{1-(i+2)t} K \int_{[0,\infty)^4} \frac{e^{-\sum_{i=1}^4 x_i(\frac{1}{i+2}-t)}}{\prod_{i=1}^4 \frac{i+2}{1-(i+2)t)}} \frac{e^{-tS(\mathbf{x})} S(\mathbf{x}) I\{S(\mathbf{x}) > 62\}}{\prod_{i=1}^4 (i+2)} \, d\mathbf{x};$$

$$\approx \sum_{j=1}^{N} S(\mathbf{X}_j) I\{S(\mathbf{X}_j) > 62\} e^{-tS(\mathbf{X}_j)} \Big/ \sum_{j=1}^{N} I\{S(\mathbf{X}_j) > 62\} e^{-tS(\mathbf{X}_j)}.$$

Text estimates "good" $t = .14$, by approximately solving

$\sum_{i=1}^4 E_t[X_i] = \frac{3}{1-3t} + \frac{4}{1-4t} + \frac{5}{1-5t} + \frac{6}{1-6t} = 62.$

Matlab tests:

```
N = 100000; U = rand(4,N);
t = .14; Cd = 1./(1-[3:6]*t); St = -([3:6].*Cd)*log(1-U);
fg = ( St > 62 ).*exp(-t*St); E = St.*fg/mean(fg);
disp([mean(E) 2*std(E)/sqrt(N)])% Expected Value
       68.236       1.0897
```

Note smaller standard errors compared to raw NIM MC.

LATIN HYPERCUBE SAMPLING

Latin Hypercube Sampling : for

$$\theta = E[h(U_1, U_2, \ldots, U_n)] = \int_0^1 \int_0^1 \cdots \int_0^1 h(x_1, x_2, \ldots, x_n) dx_1 dx_2 \cdots dx_n.$$

- Raw MC would use $U_{ik} \sim Uniform(0, 1)$, with $i = 1, \ldots, n$, $k = 1, \ldots, N$, for

$$\theta \approx \frac{1}{N} \sum_{k=1}^{N} h(U_{1k}, U_{2k}, \ldots, U_{nk}).$$

Latin Hypercube sampling "stratifies" each x_i variable: replace U_{ik} by

$$U^*_{ik} = (U_{ik} + p_{ik} - 1)/N,$$

where, for each i, $p_{i1}, p_{i2}, \ldots, p_{iN}$ is a random permutation of $1, 2, ..., N$.

Usually N is divided by K, and K subsamples are used to estimate

$$\theta \approx \frac{1}{K} \sum_{j=1}^{K} \frac{1}{M} \sum_{k=1}^{M} h(U^*_{1k}, U^*_{2k}, \ldots, U^*_{nk}).$$

with $M = \frac{N}{K}$ and $U^*_{ik} = (U_{ik}+p_{ik}-1)/M$, and p_{ik}'s are a permutation of $1, \ldots, M$

LATIN HYPERCUBE SAMPLING CONT.

- LHCS Example $\theta = \int_0^1 \int_0^1 e^{(x_1+x_2)^2} dx_1 dx_2$;

```
N = 10000; U = rand(2,N); T = exp(sum(U).^2);
disp( [mean(T) 2*std(T)/sqrt(N)]) % Raw MC
       4.9781       0.1234
K = 20; M = N/K;
for j = 1 : K, U = rand(2,M);
  for i = 1 : 2, US(i,:) = ( U(i,:) + randperm(M) - 1 )/M; end
  TL(j) = mean( exp(sum(US).^2) );
end, disp( [mean(TL) 2*std(TL)/sqrt(K)]) % Latin Hypercube MC
       4.8762      0.05574
```

LATIN HYPERCUBE SAMPLING CONT.

- Bridge Network Example: estimate expected value of length of shortest path

  $L = E[\min(X_1 + X_4, X_1 + X_3 + X_5, X_2 + X_3 + X_4, X_2 + X_5)] \equiv E[\min(P(\mathbf{X}))]$,

  with $X_i \sim Uniform(0, a_i)$, $\mathbf{a} = (1, 2, 3, 1, 2)$.

  Raw MC Matlab:

```
N = 10000; B = [1 0 0 1 0;1 0 1 0 1;0 1 1 1 0;0 1 0 0 1];
U = diag([1 2 3 1 2])*rand(5,N); L = min(B*U);
disp( [mean(L) 2*std(L)/sqrt(N)])         % Raw MC
      0.93147    0.0079154
K = 20; M = N/K;
for j = 1 : K, U = rand(5,M);
  for i = 1 : 5, US(i,:) = ( U(i,:) + randperm(M) - 1 )/M; end
  LL(j) = mean( min(B*diag([1 2 3 1 2])*US) );
end, disp( [mean(LL) 2*std(LL)/sqrt(K)]) % Latin Hypercube MC
     0.92926    0.0026256
```

  Note: exact value = .9298611111 . . .

LATIN HYPERCUBE SAMPLING CONT.

- Example: if $X_i \sim Exp(\frac{1}{i+2})$, $i = 1, 2, 3, 4$, $S(\mathbf{X}) = \sum_{i=1}^{4} X_i$, estimate

$$\theta = \int_0^\infty \cdots \int_0^\infty \frac{e^{-\sum_{i=1}^{4} \frac{x_i}{i+2}}}{3 \cdot 4 \cdot 5 \cdot 6} S(\mathbf{x}) I\{S(\mathbf{x}) > 5\}\, d\mathbf{x}.$$

Raw MC uses $X_{ij} \sim Exp(\frac{1}{i+2})$, to estimate $\theta \approx \frac{1}{N} \sum_{j=1}^{N} h(\mathbf{X}_j)$,
with $h(\mathbf{X}) = S(\mathbf{X}) I\{S(\mathbf{X}) > 5\}$.
Matlab:

```
N = 100000; U = rand(4,N);
X = -diag([3:6])*log(1-U); S = sum(X); h = S.*( S > 5 );
disp( [mean(h) 2*std(h)/sqrt(N)])         % Raw MC
      17.903     0.059999
K = 20; M = N/K;
for j = 1 : K, U = rand(4,M);
  for i = 1 : 4, US(i,:) = ( U(i,:) + randperm(M) - 1 )/M; end
  X = -diag([3:6])*log(1-US); S = sum(X);
  XL(j) = mean( S.*( S > 5 ) );
end, disp( [mean(XL) 2*std(XL)/sqrt(K)]) % Latin Hypercube MC
      17.888    0.0034571
```

LATIN HYPERCUBE SAMPLING CONT.

- Example: if $X_i \sim Exp(\frac{1}{i+2})$, $i = 1, 2, 3, 4$, $S(\mathbf{X}) = \sum_{i=1}^4 X_i$, estimate

$$\theta = K \int_0^\infty \cdots \int_0^\infty \frac{e^{-\sum_{i=1}^4 \frac{x_i}{i+2}}}{3 \cdot 4 \cdot 5 \cdot 6} S(\mathbf{x}) I\{S(\mathbf{x}) > 62\}\, d\mathbf{x},$$

with $K^{-1} = \int_0^\infty \cdots \int_0^\infty \frac{e^{-\sum_{i=1}^4 \frac{x_i}{i+2}}}{3 \cdot 4 \cdot 5 \cdot 6} I\{S(\mathbf{x}) > 62\}\, d\mathbf{x}$.

Raw NIM MC uses $X_{ij} \sim Exp(\frac{1}{i+2})$, to estimate $\theta \approx \frac{\sum_{j=1}^N S(\mathbf{X}_j) I\{S(\mathbf{X}_j) > 62\}}{\sum_{j=1}^N I\{S(\mathbf{X}_j) > 62\}}$.

Matlab:

```
N = 100000; U = rand(4,N); X = -diag([3:6])*log(1-U);
S = sum(X); h = S.*( S > 62 ); E = h/mean( S > 62 );
disp([mean(E) 2*std(E)/sqrt(N)])% Raw MC Expected Value
    68.551       15.604
K = 20; M = N/K;
for j = 1 : K, U = rand(4,M);
  for i = 1 : 4, US(i,:) = ( U(i,:) + randperm(M) - 1 )/M; end
  X = -diag([3:6])*log(1-US);
  S = sum(X); XL(j) = mean( S.*( S > 62 ) )/mean( S > 62 );
end, disp( [mean(XL) 2*std(XL)/sqrt(K)]) % Latin Hypercube MC
    68.163       4.6203
```

LATIN HYPERCUBE SAMPLING CONT.

```
N = 1000000; U = rand(4,N); X = -diag([3:6])*log(1-U);
S = sum(X); h = S.*( S > 62 ); E = h/mean( S > 62 );
disp([mean(E) 2*std(E)/sqrt(N)])% Raw MC Expected Value
    68.262       4.6398
K = 20; M = N/K;
for j = 1 : K, U = rand(4,M);
  for i = 1 : 4, US(i,:) = ( U(i,:) + randperm(M) - 1 )/M; end
  X = -diag([3:6])*log(1-US);
  S = sum(X); XL(j) = mean( S.*( S > 62 ) )/mean( S > 62 );
end, disp( [mean(XL) 2*std(XL)/sqrt(K)]) % Latin Hypercube MC
    68.071      0.36825
```

Note error estimate reduction for LHC by factor > 10 compared to $N = 100000$