



STK4051/STK9051

Computational Statistics

Riccardo De Bin

`debin@math.uio.no`

Outline of the lecture

- Simulation and Monte Carlo Integration
 - Introduction to the Monte Carlo Method
 - Exact Simulation
 - Generating from Standard Parametric Families
 - Inverse Cumulative Distribution Function
 - Rejection Sampling

Simulation and Monte Carlo Integration: Introduction

Monte Carlo Integration:

- **statistical estimation** of the value of an integral;
- using evaluations of an integrand at points **randomly drawn**;
- from a distribution with **support over the range** of integration.

Some uses:

- Likelihood analyses:
 - ▶ marginalization of a joint density;
 - ▶ Maximum likelihood fit of a generalized linear mixed model.
- Bayesian analysis:
 - ▶ computation of posterior moments;
 - ▶ computation of posterior probabilities.

Introduction to the Monte Carlo Method: general notes

Many quantities of interest in inferential statistical analyses can be expressed as the **expectation** of a function of a r.v., $E\{h(\mathbf{X})\}$.

Let

- f denote the density of $\mathbf{X}$;
- μ denote the expectation of $h(\mathbf{X})$ with respect to f .

When an i.i.d. random sample $\mathbf{X}_1, \dots, \mathbf{X}_n$ is obtained from f ,

- we can approximate μ by a sample average,

$$\hat{\mu}_{\text{MC}} = \frac{1}{n} \sum_{i=1}^n h(\mathbf{X}_i) \rightarrow \int h(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} = \mu$$

as $n \rightarrow \infty$ by the **strong law of large numbers**.

Introduction to the Monte Carlo Method: general notes

Many other quantities can be estimated in this manner.

Let

- $v(\mathbf{x}) = \{h(\mathbf{x}) - \mu\}^2$;
- assume that $h(\mathbf{X})^2$ has finite expectation under f ;
- the sampling variance of $\hat{\mu}_{\text{MC}}$ is $\sigma^2/n = \text{E}_f[v(\mathbf{X})/n]$, where the expectation is taken with respect to f ;

Monte Carlo estimator of σ^2 :

$$\widehat{\text{var}}\{\hat{\mu}_{\text{MC}}\} = \frac{1}{n-1} \sum_{i=1}^n (h(\mathbf{X}_i) - \hat{\mu}_{\text{MC}})^2$$

When σ^2 exists, by the **central limit theorem**,

$$\hat{\mu}_{\text{MC}} \overset{n \rightarrow \infty}{\rightsquigarrow} N(\mu, \sigma^2/n)$$

so statistical inference for μ follows.

Introduction to the Monte Carlo Method: general notes

Monte Carlo integration provides **slow convergence**:

- it has convergence of **order** $\mathcal{O}(n^{-1/2})$;
- quadrature methods of order $\mathcal{O}(n^{-2})$ or better.

Advantages of Monte Carlo:

- less hampered by **high dimensionality**
 - does not attempt any systematic exploration of the p -dimensional support region of f ;
- does **not** require the integrand be **smooth**.

Exact Simulation: Generating from Standard Parametric Families

Distinction between **exact** and **approximate** simulations;

- exact: from a sampling distribution that is **exactly** f ;
- approximate: from a sampling distribution that **approximate** f .

Three basic approaches for **exact simulation** of random variables:

- if F^{-1} is available, simply **generate from** $F^{-1}(U)$,
 - ▶ $\forall$ continuous F , if $U \sim \text{Unif}(0, 1)$, then $X = F^{-1}(U) = \inf\{x : F(x) \geq U\}$ has cumulative distribution function F .

Examples:

- ▶ $U \sim \text{Unif}(0, 1)$, $X = -\log(1 - U)/\lambda \rightarrow X \sim \text{Exp}(\lambda)$;
- ▶ $U_1, U_2 \sim \text{Unif}(0, 1)$, $X = \mu + \sigma\sqrt{-2\log U} \cos\{2\pi U_2\} \rightarrow X \sim N(\mu, \sigma^2)$;
- ▶ ...

Exact Simulation: Inverse Cumulative Distribution Function

- if F^{-1} is not available but F is (or easily approximated),
 - ▶ make a **grid of points** $x_1, \dots, x_m$ that cover the support of f ;
 - ▶ calculate $u_1, \dots, u_m$ as $F(x_1), \dots, F(x_m)$;
 - ▶ draw $U \sim \text{Unif}(0, 1)$ and **interpolate** between $u_i \leq U \leq u_j$,

$$X = \frac{u_j - U}{u_j - u_i} x_i + \frac{U - u_i}{u_j - u_i} x_j.$$

Note: this approach is not exact, but the **approximation is deterministic** and can be made **arbitrarily small** (increasing m).

- **rejection sampling.**

Rejection Sampling: Algorithm

Goal: generate **realizations from target density** f .

- $f(x)$ can be computed (at least up to a constant);
- g is a density we can (i) **sample from**, (ii) **easily calculate** $g(x)$;
- let $e(\cdot)$ denote an **envelope**,
 - ▶ given a constant $\alpha \leq 1$, $e(x) = g(x)/\alpha \geq f(x)$, $\forall x : f(x) > 0$.

Rejection sampling algorithm:

1. sample $Y \sim g$;
2. sample $U \sim \text{Unif}(0, 1)$;
3. if $U > f(Y)/e(Y)$ **reject** Y ;
4. if $U \leq f(Y)/e(Y)$, **keep** Y , i.e., set $X = Y$ and consider X to be an element of the target random sample;

Repeat 1–3 until reaching a sample of the desired size.

Rejection Sampling: Exactness

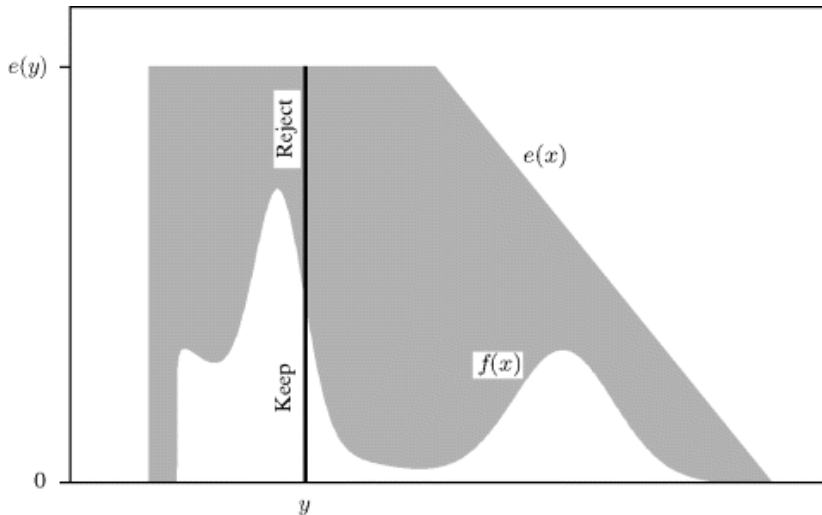
The draws are an **independent sample** from f , **no approximation** involved,

$$\begin{aligned} P[X \leq y] &= P \left[Y \leq y \mid U \leq \frac{f(Y)}{e(Y)} \right] \\ &= P \left[Y \leq y \text{ and } U \leq \frac{f(Y)}{e(Y)} \right] / P \left[U \leq \frac{f(Y)}{e(Y)} \right] \\ &= \int_{-\infty}^y g(z) \int_0^{f(z)/e(z)} du \, dz / \int_{-\infty}^{\infty} g(z) \int_0^{f(z)/e(z)} du \, dz \\ &= \int_{-\infty}^y f(z) dz = F(y) \end{aligned}$$

Therefore:

- the sampling distribution is **exact**;
- α can be seen as the **expected proportion of acceptance**;
- hence α is a measure of the **efficiency** of the algorithm.

Rejection Sampling: Visualization



(Givens & Hoeting, 2013, figure 6.1)

Rejection Sampling: Visual Explanation

Consider Figure 6.1:

- sampling $U \sim \text{Unif}(0, 1)$ and rejecting if $U > f(y)/e(y)$ **same as** sampling $U|y \sim \text{Unif}(0, e(y))$ and keep y if $U < f(y)$;
- suppose y falls at the point indicated by the vertical bar;
- **sample** $U|Y = y$ uniformly along the vertical bar;
- rejection rule eliminates Y with probability **proportional** to the length of the bar above $f(y)$ relative to the overall bar length;
- we basically **sample** from region **under the curve e** and **throw away** points falling **above f and below e** ;
- rejection sampling provides draws **exactly** from f .

Rejection Sampling: f known up to a constant

Suppose that f is only known up to a proportionality constant c ,

- we can only compute easily $q(x) = f(x)/c$, with c unknown;
- e.g., Bayesian inference when f is the product of the prior and the likelihood scaled by some normalizing constant;

Rejection sampling can be applied in such cases:

- find an envelope e such that $e(x) \geq q(x)$ for all $x : q(x) > 0$;
- a draw $Y = y$ is rejected when $U > q(y)/e(y)$;
- the sampling probability remains correct,
 - ▶ the unknown constant c cancels out in the numerator and denominator when f is replaced by q ;
- the proportion of kept candidates is α/c .

Rejection Sampling: rejection sampling envelopes

Main challenge: construct a **suitable envelope**.

A **good** rejection sampling **envelopes** have three properties:

- they **exceed the target** everywhere;
- they are **easy to sample**;
- they generate **few rejections**.

In particular:

- we must **know enough** about the target to bound it;
- may require **optimization** or a **clever approximation** to f or q ;
- see soon: Adaptive Rejection Sampling.

Rejection Sampling: Example

Consider the problem of **generating a Gamma($r, 1$)** when $r \geq 1$.

- wish to generate from the density that is proportional to
$$q(y) = \exp\{a \log\{t(y)/a\} - t(y) + a\},$$
with $t(y) = a(1 + by)^3$ for $-\frac{1}{b} < y < \infty$, $a = r - \frac{1}{3}$, $b = \frac{1}{\sqrt{9a}}$.

Since q fits snugly under the function $e(y) = \exp\{-y^2/2\}$, rejection sampling amounts to:

- sampling a standard normal random variable, Z ;
- sampling a standard uniform random variable, U ;
- setting $X = t(Z)$ (accepting) if $t(Z) > 0$ and

$$U \leq \frac{q(Z)}{e(Z)} = \exp\left\{\frac{Z^2}{2} + a \log\left\{\frac{t(Z)}{a}\right\} - t(Z) + a\right\}$$

- otherwise, the candidate point is rejected;
- when $r = 4$, over **99%** of candidate draws are **accepted**.

Rejection Sampling: Example

Suppose:

- 10 obs (8, 3, 4, 3, 1, 7, 2, 6, 2, 7) from i.i.d. $X_i|\lambda \sim \text{Poisson}(\lambda)$;
- a lognormal prior for λ : $\log \lambda \sim N(\log 4, 0.5^2)$;

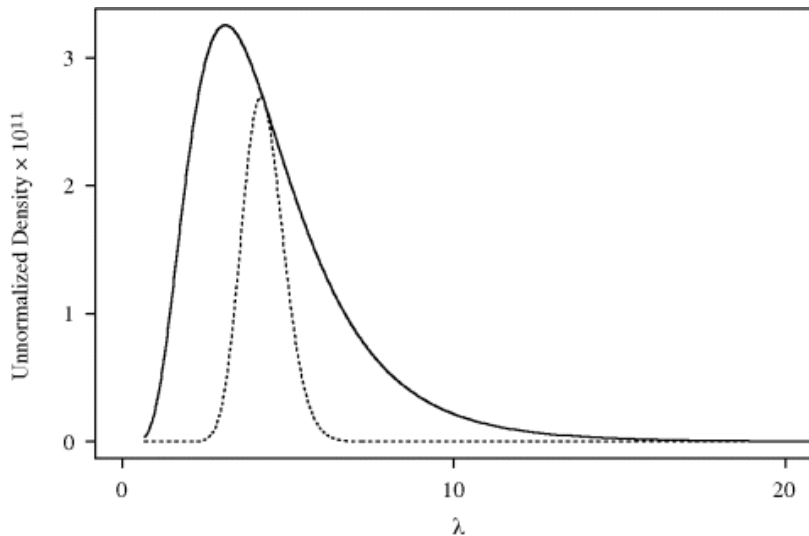
Denote the likelihood as $L(\lambda|x)$ and the prior as $f(\lambda)$:

- $\hat{\lambda} = \bar{x} = 4.3$ maximizes $L(\lambda|x)$ with respect to λ ;
- therefore the **unnormalized posterior**, $q(\lambda|x) = f(\lambda)L(\lambda|x)$ is bounded above by $e(\lambda) = f(\lambda)L(4.3|x)$.

The rejection sampling:

- sample λ_i from the lognormal prior;
- sample U_i from a standard uniform distribution;
- λ_i is kept if $U_i < q(\lambda_i|x)/e(\lambda_i) = L(\lambda_i|x)/L(4.3|x)$;
- otherwise, λ_i is rejected and the process is begun anew;
- **not super efficient** – only about 30% of candidates are kept;
- the approach is **easy and exact**.

Rejection Sampling: Example



(Givens & Hoeting, 2013, figure 6.2)

Rejection Sampling: Squeezed Rejection Sampling

Squeezed rejection sampling aims at speed up the algorithm:

- in the case evaluating f is **computationally expensive**;
- rejection sampling is otherwise **appealing**.

Idea:

- **pre-empt** the evaluation of f in some instances by employing a non-negative **squeezing function** s ;
- $s(x)$ must **not exceed** $f(x)$ anywhere on the support of f ;
- an envelope, e , is **also used**,
 - ▶ as with ordinary rejection sampling, $e(x) = g(x)/\alpha \geq f(x)$ on the support of f .

Rejection Sampling: Squeezed Rejection Sampling

The algorithm proceeds as follows:

1. sample $Y \sim g$;
2. sample $U \sim \text{Unif}(0, 1)$;
3. if $U \leq s(Y)/e(Y)$, keep the value of Y and go to step 6;
4. otherwise, if $U \leq f(Y)/e(Y)$, keep Y and go to step 6;
5. If Y has not yet been kept, reject it;
6. return to step 1 until obtaining the desired size.

Note:

- Y is kept with $p = \frac{f(y)}{e(y)}$, rejected with $p = \frac{[e(y)-f(y)]}{e(y)}$;
- same probabilities as with simple rejection sampling;
- some decisions made based on s rather than f ;
- best when s nestles up just underneath f everywhere.

Rejection Sampling: Squeezed Rejection Sampling

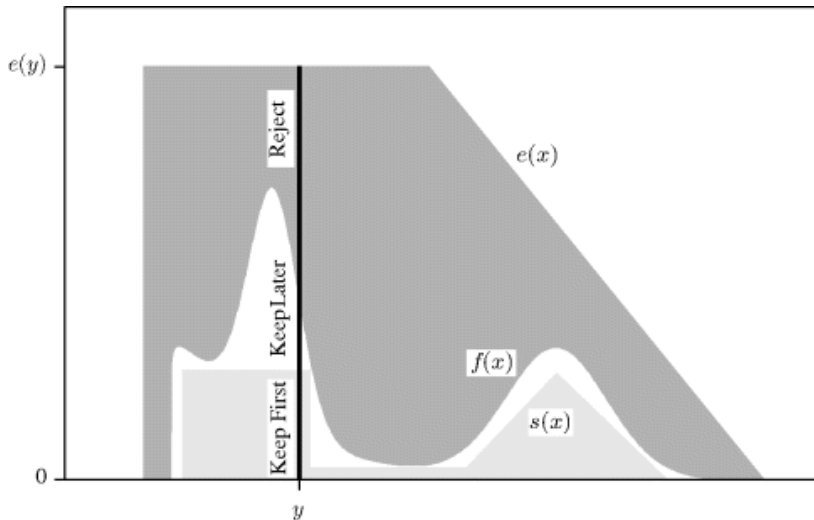
Visual explanation:

- given y , **equivalent** to sampling from $U \sim \text{Unif}(0, e(y))$;
- if $U \leq s(y)$, the candidate is **kept immediately**;
- if y not immediately kept, **second** test to check if $U \leq f(y)$;
- otherwise rejected.

Note:

- as rejection sampling, the proportion of candidates kept is α ;
- proportion of iterations **without** evaluation of f : $\frac{\int s(x)dx}{\int e(x)dx}$;
- squeezed rejection sampling can also be carried out when the target is known only **up to** a proportionality **constant**;
- the method is **still exact**, same efficiency considerations;
- generalizations for multivariate targets are straightforward.

Rejection Sampling: Squeezed Rejection Sampling



(Givens & Hoeting, 2013, figure 6.3)

Rejection Sampling: Adaptive Rejection Sampling

Adaptive rejection sampling:

- address the **most challenging** aspect of the rejection sampling strategy, the construction of a **suitable envelope**;
- for squeezed rejection sampling for a **continuous, differentiable, log-concave density** on a connected support;
- the envelope and squeezing function are **iteratively refined** concurrently with the generation of sample draws;
- The amount of waste and the frequency with which f must be evaluated both **shrink** as iterations increase.

Rejection Sampling: Adaptive Rejection Sampling

Let:

- $\ell(x) = \log f(x)$, with $f(x) > 0$ on a (possibly infinite) interval of R ;
- f be **log-concave**, i.e. $\ell(a) - 2\ell(b) + \ell(c) < 0 \ \forall a < b < c$ in the support region of f ;
- assuming f continuous and differentiable, $\ell'(x)$ exists.

Initialization of the algorithm:

- define $T_k = \{x_1, \dots, x_k\}$, $x_1 < x_2 < \dots < x_k$;
- evaluate ℓ and ℓ' at these **k points**;
- if the support of f includes $-\infty$, choose $x_1 : \ell'(x_1) > 0$;
- similarly if the support of f includes ∞ , $x_n : \ell'(x_k) < 0$.

Rejection Sampling: Adaptive Rejection Sampling

Define the **rejection envelope** on T_k to be the exp of the **piecewise linear upper hull** of ℓ formed by the **tangents** to ℓ at each $x_k \in T_k$,

$$e_k(x) = \exp\{e_k^*(x)\}$$

with $e_k^*(x)$ being the upper hull of ℓ .

One can show that the tangents at x_i and x_{i+1} **intersect at**

$$z_i = \frac{\ell(x_{i+1}) - \ell(x_i) - x_{i+1}\ell'(x_{i+1}) + x_i\ell'(x_i)}{\ell'(x_i) - \ell'(x_{i+1})}$$

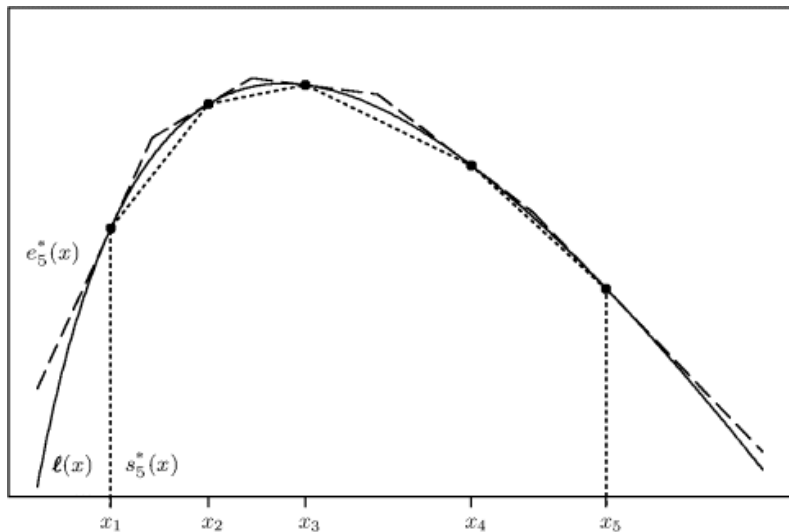
for $i = 1, \dots, k-1$.

Therefore,

$$e_k^*(x) = \ell(x_i) + (x - x_i)\ell'(x_i) \text{ for } x \in [z_{i-1}, z_i]$$

and $i = 1, \dots, k$, with z_0 and z_k defined to equal the (possibly infinite) **lower and upper bounds** of the support region for f .

Rejection Sampling: Adaptive Rejection Sampling



(Givens & Hoeting, 2013, figure 6.4)

Rejection Sampling: Adaptive Rejection Sampling

Define the **squeezing function** on T_k to be the exp of the **piecewise linear lower hull** of ℓ formed by the **chords** between adjacent $x \in T_k$.

This lower hull is given by

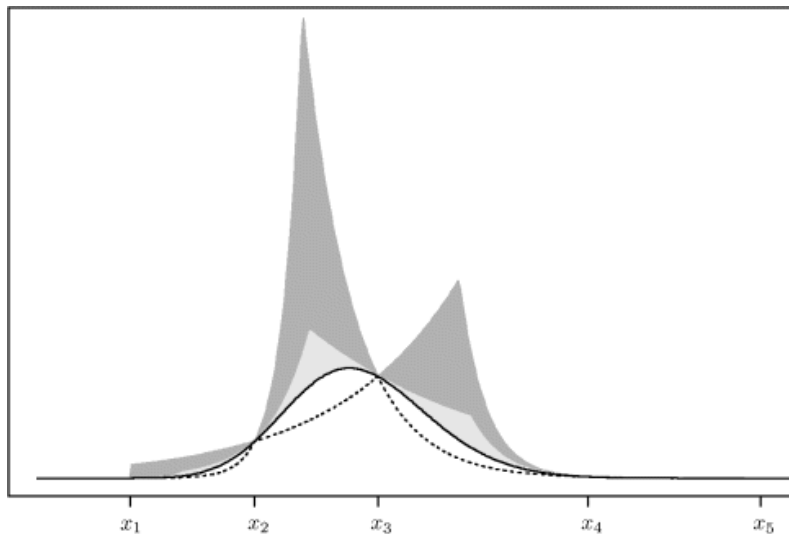
$$s_k^*(x) = \frac{(x_{i+1} - x_i)\ell(x_i) + (x - x_i)\ell(x_{i+1})}{x_{i+1} - x_i} \text{ for } x \in [x_i, x_{i+1}]$$

and $i = 1, \dots, k-1$. When $x < x_1$ or $x > x_k$, let $s_k^*(x) = -\infty$. Thus the squeezing function is $s_k^*(x)$.

Figures 6.4 and 6.5 illustrate important features of the approach:

- both $e_k(x)$ and $s_k(x)$ are **piecewise exponential functions**;
- $e_k(x)$ has exponential tails that lie **above** the tails of f ;
- $s_k(x)$ has **bounded** support.

Rejection Sampling: Adaptive Rejection Sampling



(Givens & Hoeting, 2013, figure 6.5)

Rejection Sampling: Adaptive Rejection Sampling

Adaptive rejection sampling algorithm:

- initialize by choosing a **modest** k and a **suitable grid** T_k ;
- first iteration as for squeezed rejection sampling,
- using e_k and s_k as the **envelope** and **squeezing** function;
- a candidate may be accepted without evaluating ℓ and ℓ' ;
- when a candidate is **accepted at the second stage**,
 - ▶ i.e., evaluating ℓ and ℓ' ;
 - ▶ the accepted point is **added to** T_k , creating T_{k+1} ;
 - ▶ **updated functions** e_{k+1} and s_{k+1} are also calculated;
 - ▶ then iterations **continue**;
- when a candidate is **rejected**, no update to T_k , e_k , or s_k .

Rejection Sampling: Adaptive Rejection Sampling

Note:

- candidates are taken from the density obtained by **scaling** the piecewise exponential envelope e_k so that it integrates to 1;
- since each accepted draw is made using rejection sampling, the draws are an **independent** sample **precisely from** f ;
- if f is known **only up to a multiplicative constant**,
 - ▶ the adaptive rejection sampling approach may **still be used**,
 - ▶ the proportionality constant **merely shifts** ℓ , e_k^* , and s_k^* ;
- we may need to evaluate ℓ' .

Rejection Sampling: Adaptive Rejection Sampling

Similar approach that does **not require** evaluation of ℓ' :

- assumptions and setup as the tangent-based approach above;
- for $x_i \in T_k$, define $L_i(\cdot)$ to be the **straight line** function connecting $(x_i, \ell(x_i))$ and $(x_{i+1}, \ell(x_{i+1}))$, for $i = 1, \dots, k-1$;
- define

$$e_k^*(x) = \begin{cases} \min\{L_{i-1}, L_{i+1}\} & \text{for } x \in [x_i, x_{i+1}] \\ L_1 & \text{for } x < x_1 \\ L_{k-1} & \text{for } x > x_k \end{cases}$$

with the convention that $L_0(x) = L_k(x) = \infty$;

- $e_k^*(x)$ is a piecewise linear upper hull for ℓ ,
 - ▶ the **concavity** of ℓ ensures that $L_i(x) < \ell(x)$ on (x_i, x_{i+1}) ;
 - ▶ and that $L_i(x) < \ell(x)$ when $x < x_i$ or $x > x_{i+1}$.

Rejection Sampling: Adaptive Rejection Sampling

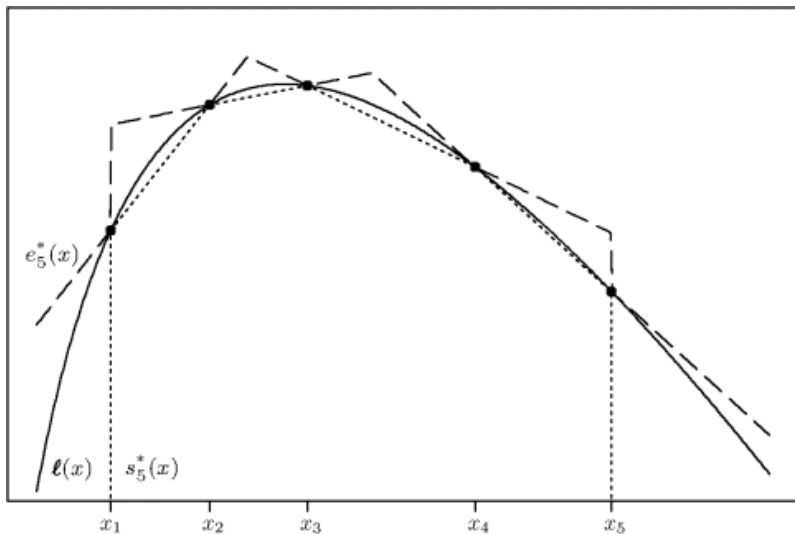
The derivative-free adaptive rejection sampling:

- the rejection sampling **envelope** is then $e_k(x) = \exp\{e_k^*(x)\}$.
- the squeezing function remains **as before**;
- iterations of the derivative-free adaptive rejection sampling algorithm proceed **analogously** to the previous approach,
 - ▶ T_k , e_k , and s_k updated each time a new point is kept.

Note:

- the envelope is **not as efficient** as when ℓ' is used;
- see the shaded area in Figure 6.5.

Rejection Sampling: Adaptive Rejection Sampling



(Givens & Hoeting, 2013, figure 6.6)

Rejection Sampling: Adaptive Rejection Sampling

Regardless of the method used to construct e_k ,

- preferably the T_k grid is **more dense** where $f(x)$ is **larger**;
- it happens **automatically**, as such points are **more likely to be kept** in subsequent iterations and included in updates to T_k ;
- grid points too far in the tails of f are **not very helpful**;
- adaptive rejection sampling can also be **extended** to densities that are not log-concave.

References |

GIVENS, G. H. & HOETING, J. A. (2013). *Computational Statistics*. Wiley.