Confidence Intervals

What is $P(\hat{\theta} = \theta)$? It's 0, because $\hat{\theta}$ is a continuous random variable if the samples are continuous r.v.'s.

Next best hope: find minimum Δ such that

$$P(\theta \in [\hat{\theta} - \Delta, \hat{\theta} + \Delta]) \geq 0.95, \text{ for instance}$$

This is the 95% confidence interval.

Ex: MLE of normal mean μ.

Assume $x_1, x_2, \ldots, x_n \sim N(\mu, \sigma^2)$ independent, where μ is unknown and σ^2 is known.

MLE $\hat{\theta}_1$ of μ is $\hat{\theta}_1 = \frac{1}{n} \sum_{i=1}^{n} x_i$.

$\hat{\theta}_1$ is a r.v. The Central Limit Theorem says $\hat{\theta}_1$ is close to normal when n is large.

$$\hat{\theta}_1 \approx N\left(\mu, \frac{\sigma^2}{n}\right).$$

$$\frac{\hat{\theta}_1 - \mu}{\sigma/\sqrt{n}} \approx N(0,1)$$

$$P\left(-z \leq \frac{\hat{\theta}_1 - \mu}{\sigma/\sqrt{n}} \leq z\right) \approx 2\Phi(z) - 1$$

$$P\left(-z \leq \frac{\mu - \hat{\theta}_1}{\sigma/\sqrt{n}} \leq z\right) \approx 2\Phi(z) - 1$$

$$P\left(\hat{\theta}_1 - z\sigma/\sqrt{n} \leq \mu \leq \hat{\theta}_1 + z\sigma/\sqrt{n}\right) \approx 2\Phi(z) - 1 = 0.95$$

$$\Phi(z) = 0.975 \Rightarrow z \approx 1.96$$

If we choose $\Delta = z\sigma/\sqrt{n} = 1.96\sigma/\sqrt{n}$,
then $P(\mu \in [\hat{\theta}_1 - \Delta, \hat{\theta}_1 + \Delta]) \approx 0.95$

Probalistic (or randomized) Algorithms.

How can we factor large integers (say, 1000 digits) quickly? Wide open problem. What about the simpler problem of determining whether this large integer x is prime or composite? This is the Primality problem.

1977: Solovay & Strassen showed how to determine primality of x in time polynomial in n (the number of digits of x) using random numbers.

Catch: with tiny probability, the algorithm says x is prime even though x is composite.

What other problems can be solved efficiently using randomness?

Quicksort (Hoare, 1959)

To sort $a_1, a_2, \ldots, a_n$: If $n > 1$,

1. Choose $p \sim \mathrm{Unif}(1, n)$.
2. Let $L = \{a_i \mid a_i < a_p\}$
   $E = \{a_i \mid a_i = a_p\}$
   $G = \{a_i \mid a_i > a_p\}$.
3. Recursively sort and output L.
   Output E.
   Recursively sort and output G.

If you're unlucky (recursively), then the running time is $\sim n^2$.