

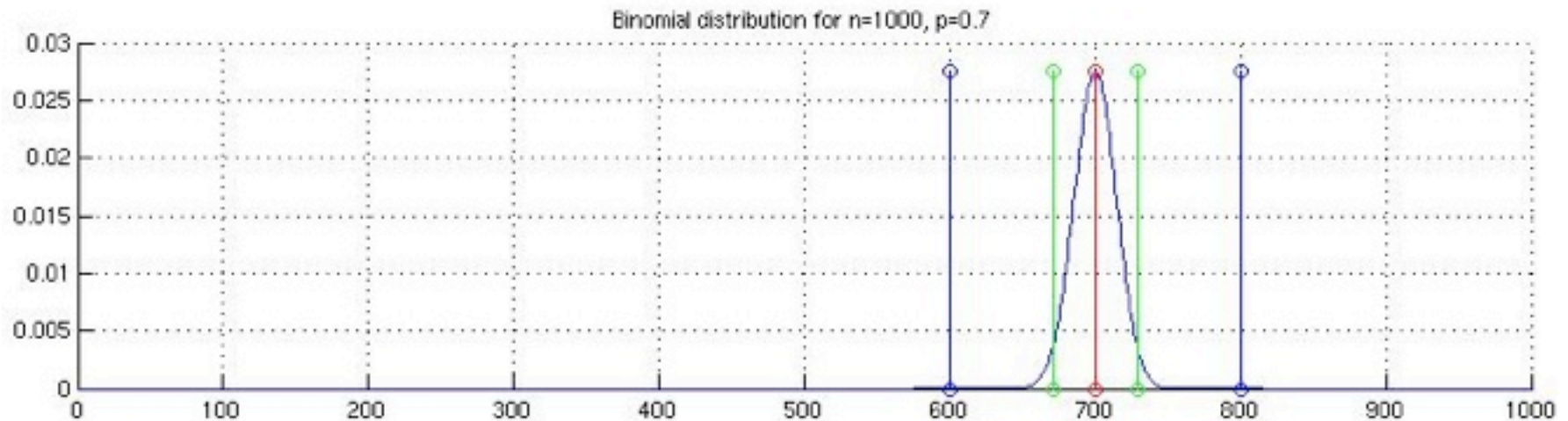
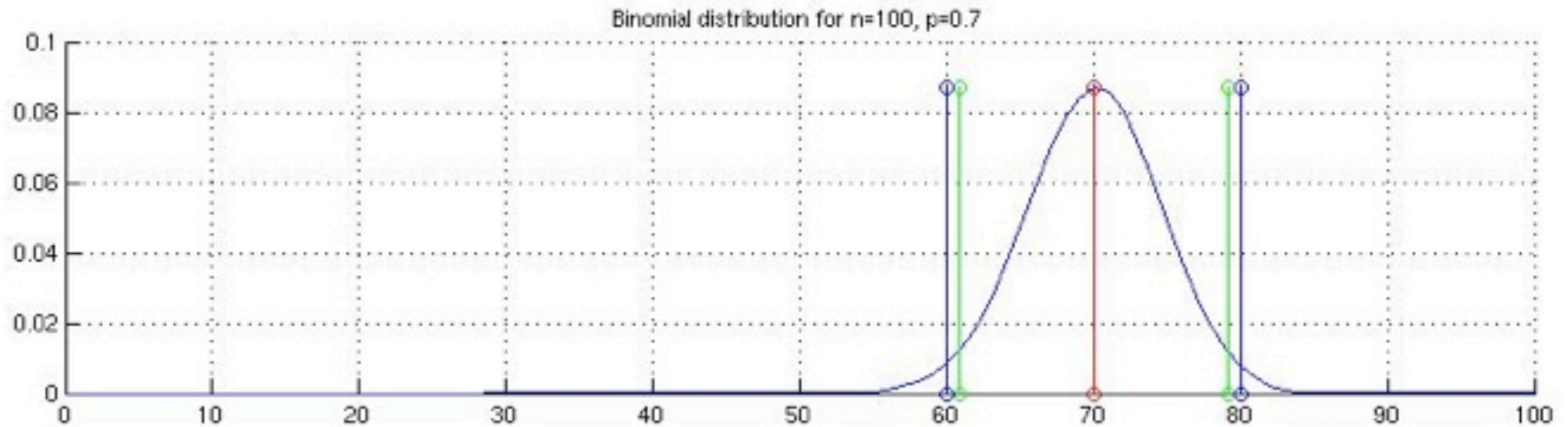
Large Deviations vs. Small Deviations

Sum of IID binary random variable

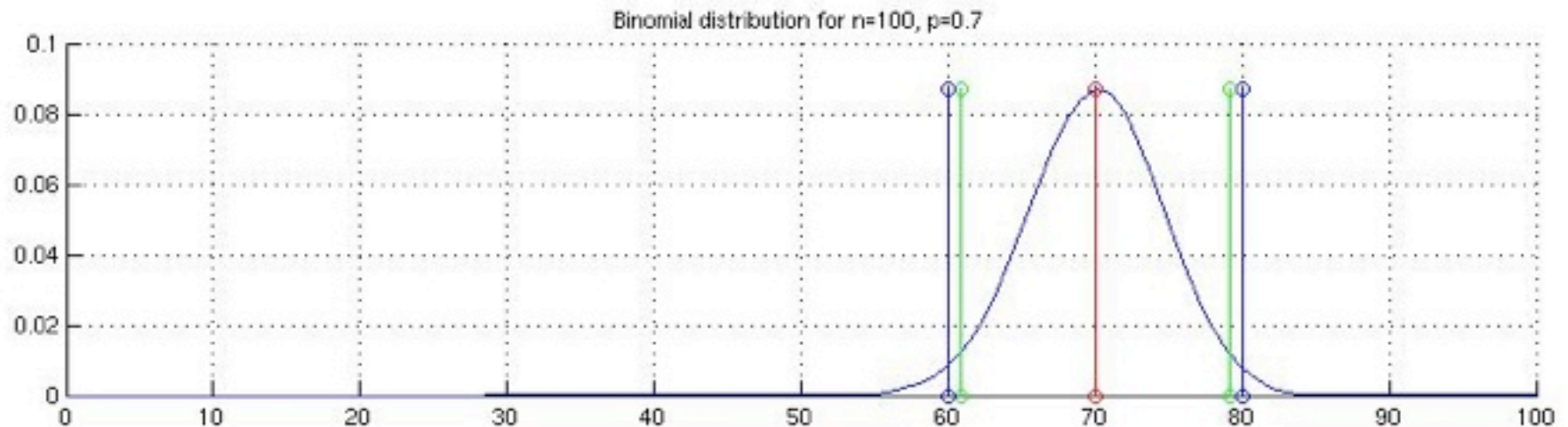
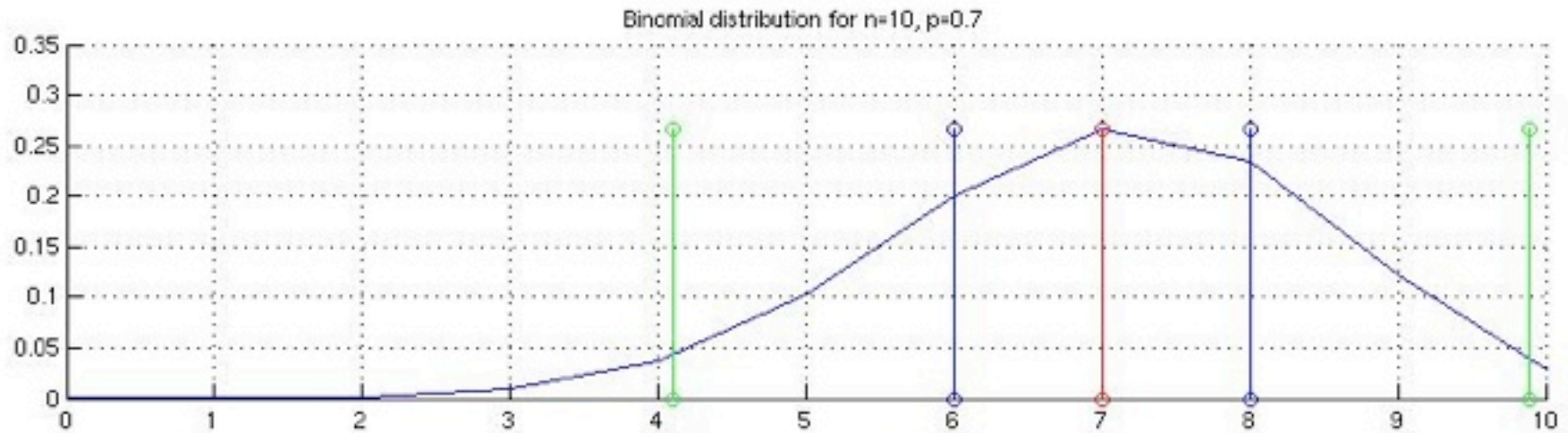
$$X_i = \begin{cases} 1 & \text{with prob. } p \\ 0 & \text{with prob. } 1 - p \end{cases}$$

$$S_n = \sum_{i=1}^n X_i$$

Small deviations vs. large deviations



Small deviations vs. large deviations



Large and small deviations are incompatible

	Scale	Prob. limit
Small Deviation analysis	$S_m/\sqrt{m}$	const
Large deviation analysis	S_m/m	e^{-cm}

Hoeffding / Chernoff / Sanov's bound

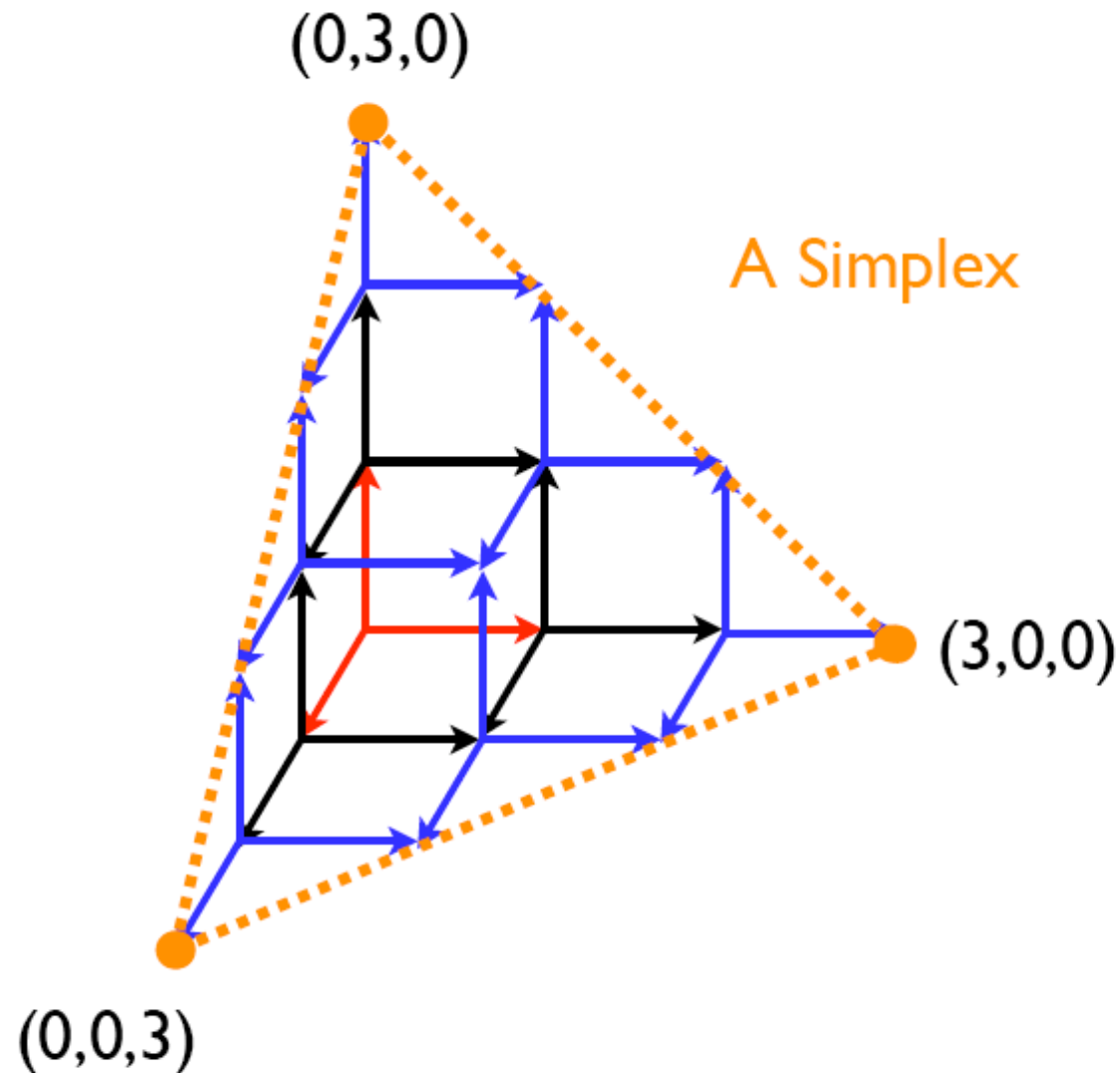
$$X_i = \begin{cases} 1, & \text{with probability } p \\ 0, & \text{with probability } 1 - p \end{cases} \quad \hat{p} = \frac{1}{m} \sum_{i=1}^m X_i$$

Our goal is to show that, with high probability, this estimate ($\hat{p}$) is pretty good. Specifically, we want to upper bound the probability $P[\hat{p} \geq q]$ for any $q > p$.

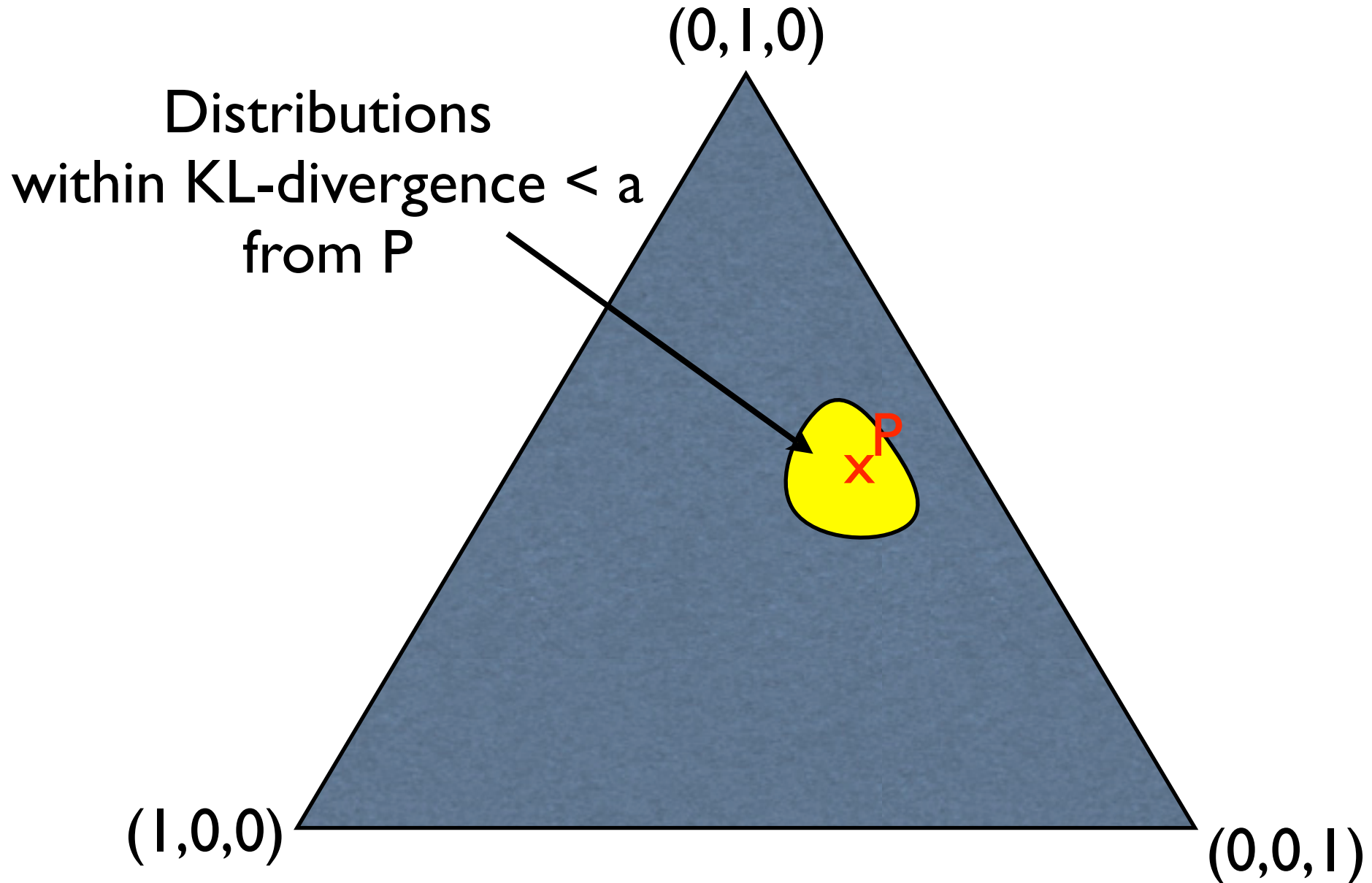
$$P[\hat{p} > q] \leq e^{-m D_{\text{KL}}(q||p)}$$

$$D_{\text{KL}}(q||p) = \sum_i q_i \log \frac{q_i}{p_i}$$

3-element distributions and the 2D simplex



Sanov's theorem



On the virtues of the cumulative distribution function

matlab code

```
function [f,x] = cdist(data)
```

```
x=sort(data);
```

```
f=[0, 0:(1/length(x)):1];
```

```
x=[x(1); x(:); x(end)];
```

CDFs and sorting

- To Create a CDF you need to sort
- Can knowing the CDF help you sort?

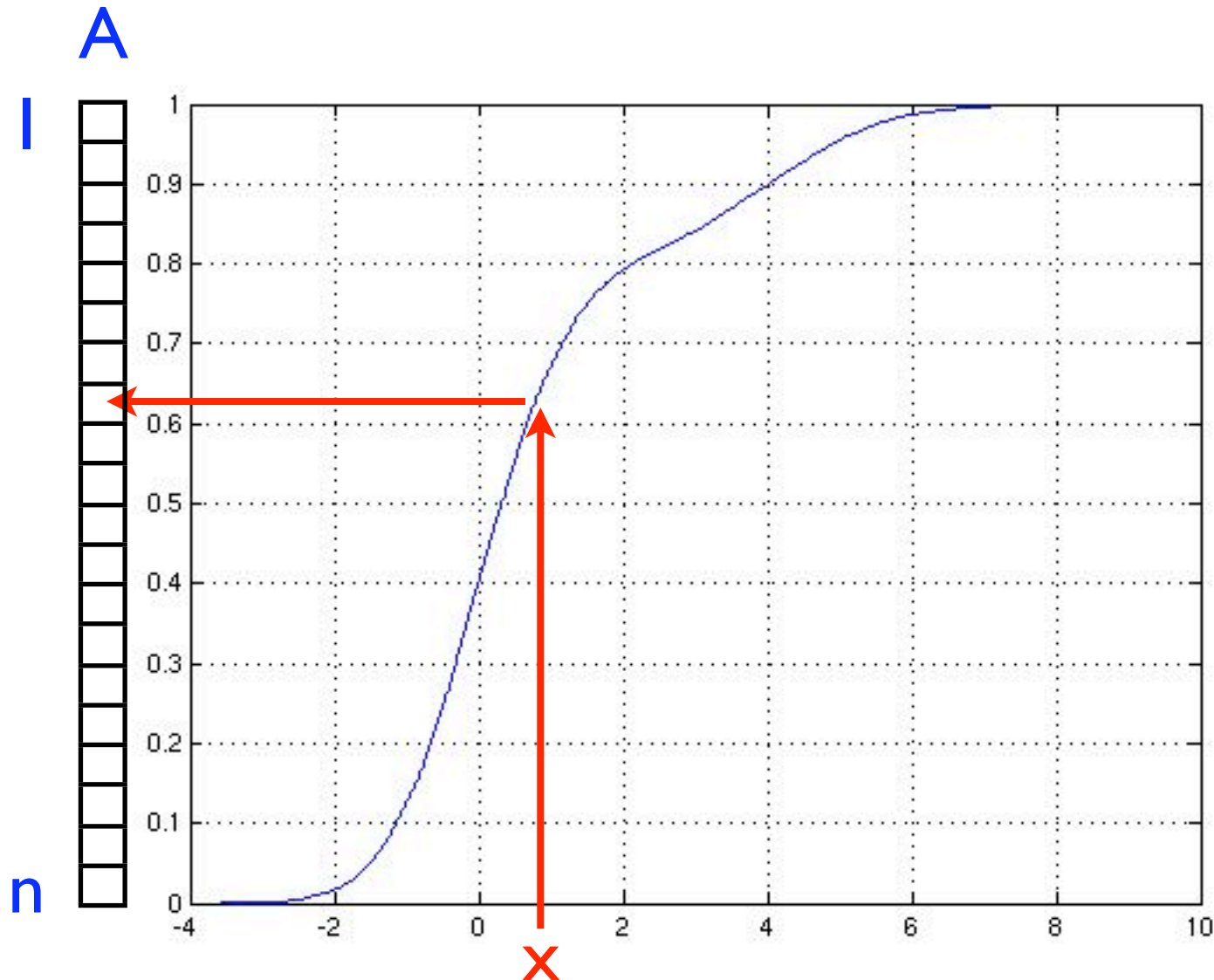
No. of comparisons needed for sorting

- There are $n!$ permutations
- Identifying the correct permutation requires $\log(n!)$ comparisons.
- $\log(n!) \sim n \log n$
- Merge-sort achieves time $O(n \log n)$

Efficient sorting for uniform distribution

- n Elements drawn IID from uniform distribution over $[0, 1]$
- $A = \text{array}[1:n]$ of lists
- given x insert it into list at $A[\text{floor}(x \cdot n)]$
- Lists would rarely have > 1 element
- time: $O(n)$

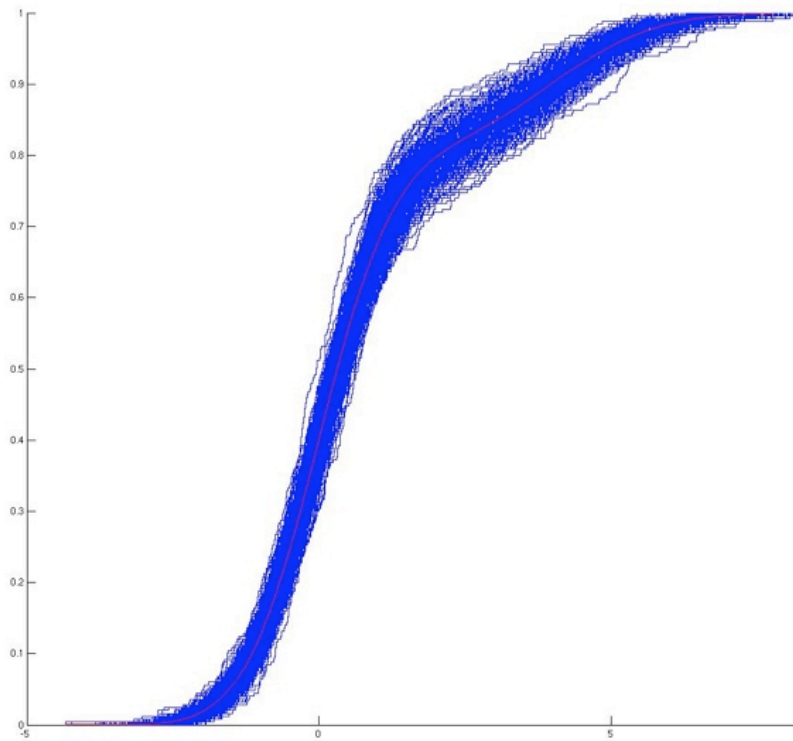
Efficient sorting using the CDF



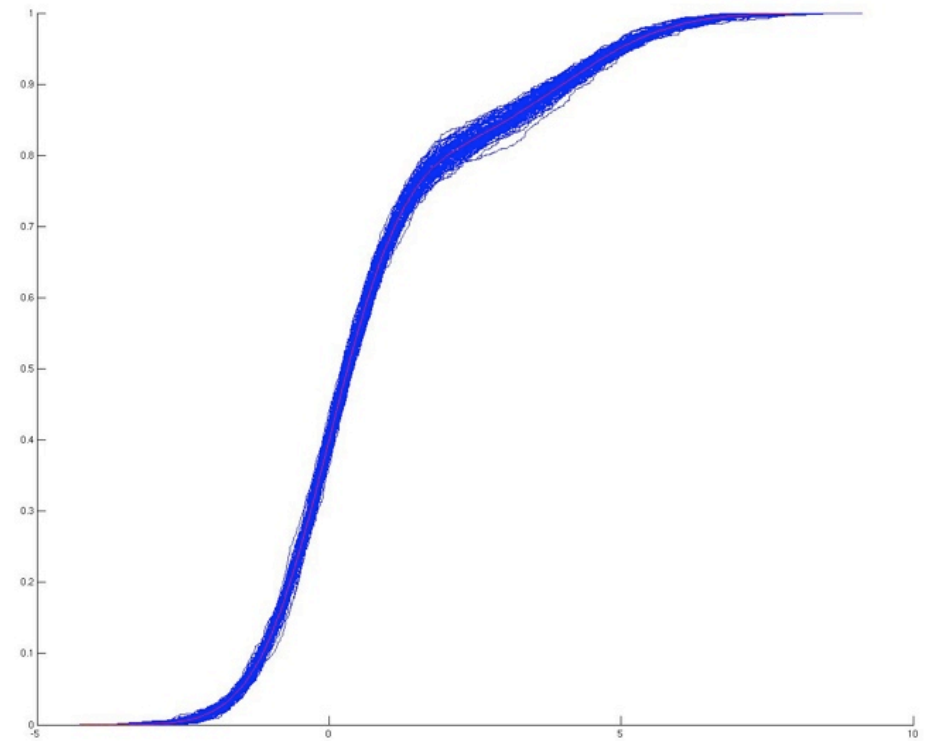
Efficient sorting for an IID source

- Source is IID but distribution is unknown
- Sort first m instances to estimate CDF
- Use CDF to sort rest of data.
- How large should m be?

Stability of empirical CDF



250 instances



1000 instances

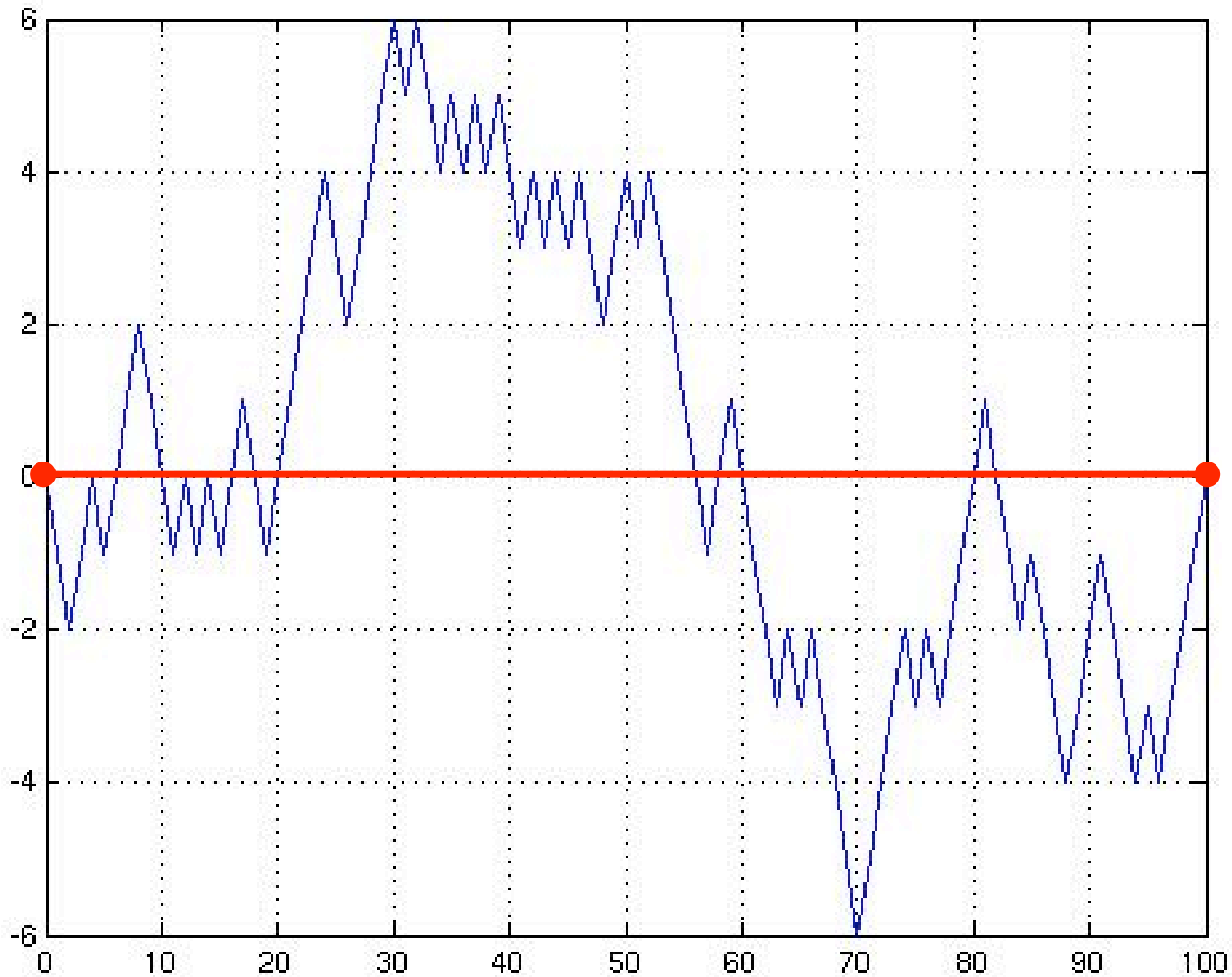
The difference between two empirical CDFs

```
% input: samples d1,d2
n = length(d1) % assuming length(d1)==length(d2)
vals = [d1; d2];
labels = [repmat(1,n,1); repmat(-1,n,1)];

[s,i] = sort(vals); % sort labels by
s_labels = labels(i); % increasing vals

d = cumsum(s_labels);
plot(0:length(d),[0;d]);
```

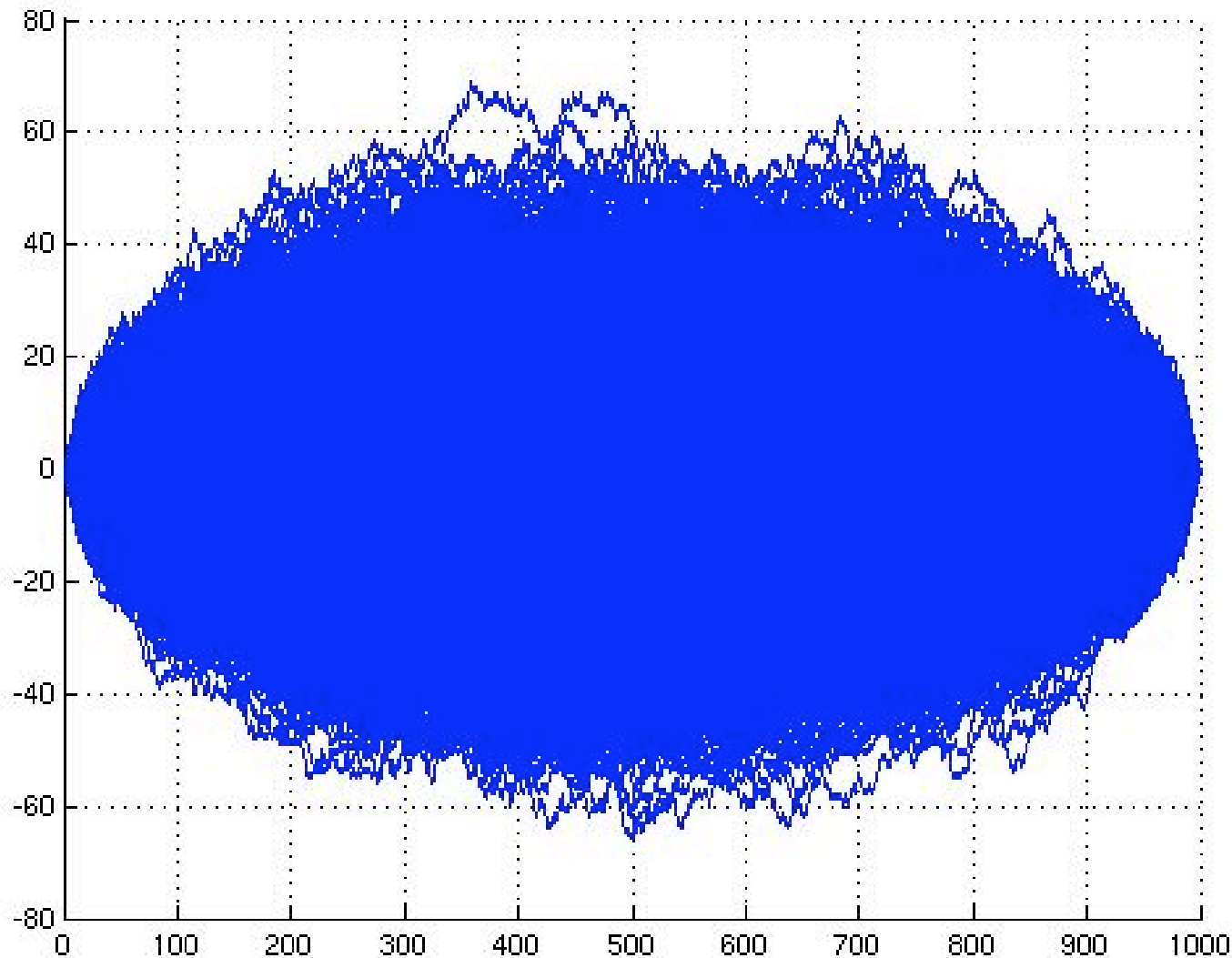
Typical cumulative difference



Typical cumulative difference



Maximal divergence of a returning random walk



The Glivenko-Cantelli Theorem

$F(x)$ = CDF

$F_n(x)$ = empirical CDF using n random instances

ϵ = error tolerance

$\mathbf{x}_1^n = \langle x_1, x_2, \dots, x_n \rangle$ = sample

$$P_{\mathbf{x}_1^n} \left(\max_x |F_n(x) - F(x)| \geq \epsilon \right) \leq a e^{-b n \epsilon^2}$$

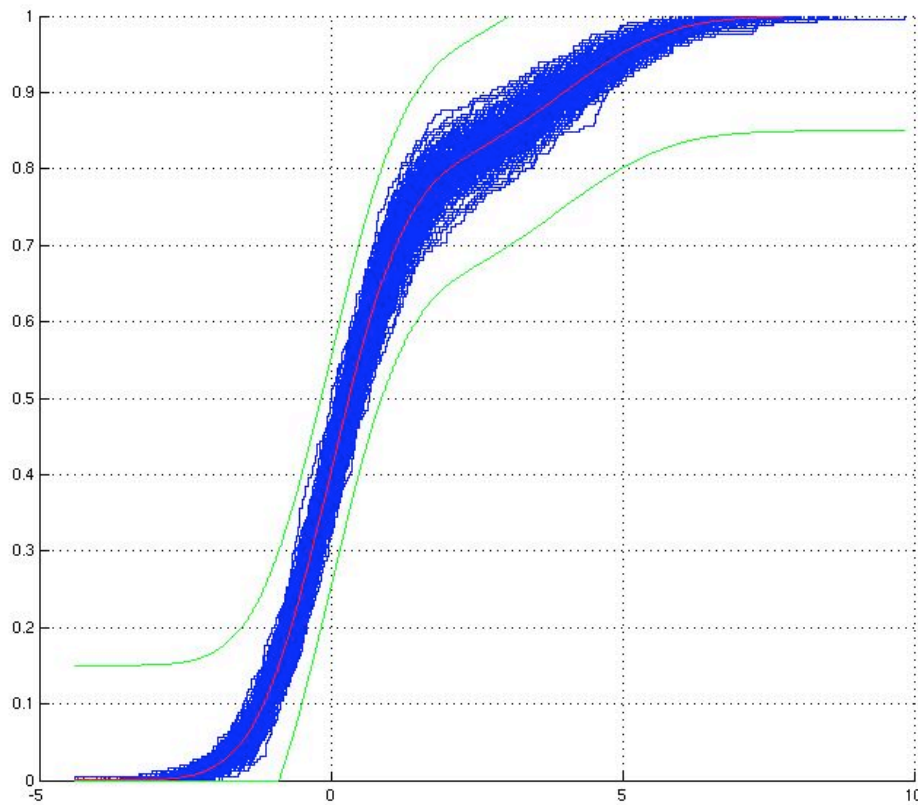
Elegant proof.

Devroye, Györfi, Lugosi / A probabilistic Theory of Pattern Recognition, page 192

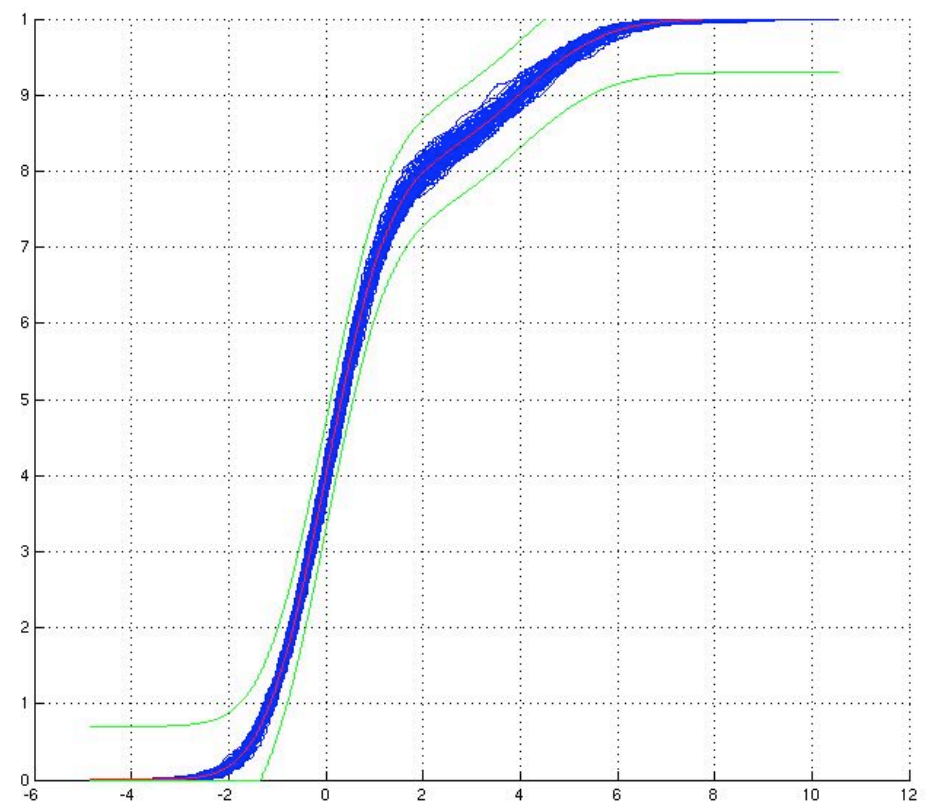
Best Constants: $a=b=2$, Complex proof.

MASSART, P. (1990). The tight constant in the Dvoretzky-Kiefer-Wolfowitz inequality. Ann. of Probability 18, 1269-1293.

Empirical test of the Glivenko-Cantelli Theorem



250 instances



1000 instances