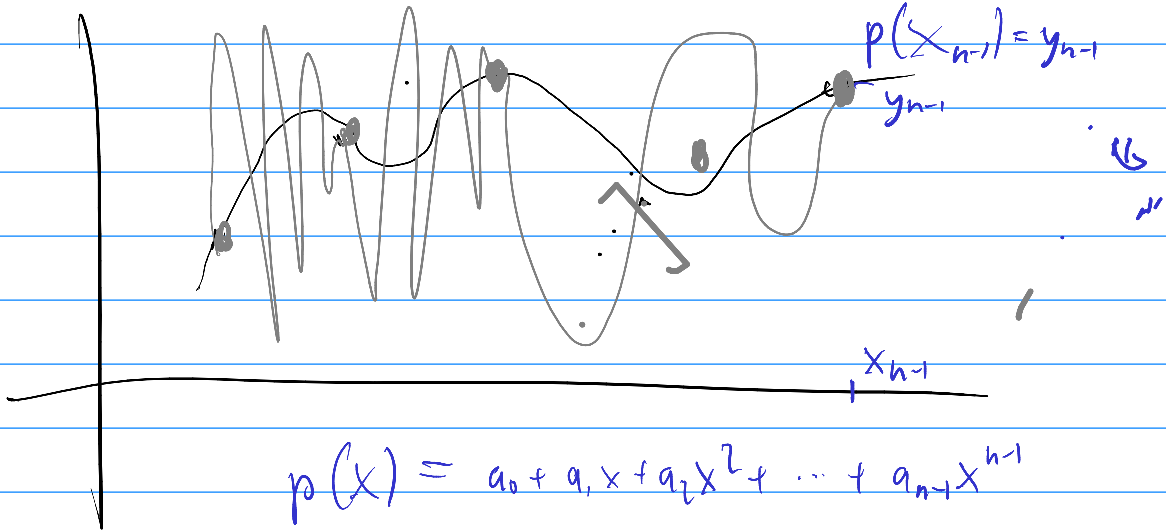


$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$$

$$p(x) = a_0 \cdot 1 + a_1 \sin(x) + a_2 \cos(x) + a_3 \sin(2x) + a_4 \cdot \cos(4x) \;\cdot\;\cdot$$

$\uparrow$
$\cos(0x)$

$$\alpha_{440} \sin(440 \cdot 2\pi \cdot t)$$

$$x^{440}$$

More General Functions

- Is this technique limited to the monomials $\{1, x, x^2, x^3, \ldots\}$?

Interpolation with General Sets of Functions

For a general set of functions $\{\varphi_1, \ldots, \varphi_n\}$, solve the linear system with the generalized Vandermonde matrix for the coefficients $(a_1, \ldots, a_n)$:

$$\underbrace{\begin{pmatrix} \varphi_1(x_1) & \varphi_2(x_1) & \cdots & \varphi_n(x_1) \\ \varphi_1(x_2) & \varphi_2(x_2) & \cdots & \varphi_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(x_n) & \varphi_2(x_n) & \cdots & \varphi_n(x_n) \end{pmatrix}}_{V} \underbrace{\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}}_{\boldsymbol{a}} = \underbrace{\begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{pmatrix}}_{\boldsymbol{f}}.$$

- Given those coefficients, what is the interpolant $\tilde{f}$ satisfying $\tilde{f}(x_i) = f(x_i)$?

3 Making Models with Monte Carlo

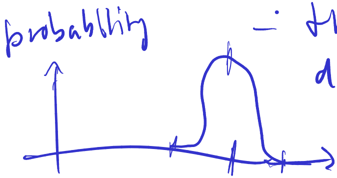
Randomness: Why?

- What types of problems can we solve with the help of random numbers?
  - Stock/option pricing
  - Robotics/mapping
  - PageRank

Random Variables

- What is a random variable?

- Depends on the (unknown) state of the world
- gives you some numerical result
- the values of the RV have a probability distribution associated with them



Discrete

$X = x_1$ $\quad$ $X = x_2$ $\quad$ $X = x_3$ ...

p_1 $\quad$ p_2 $\quad$ p_3

Assume $p_i \geq 0$

Continuous

0 ———— ( ) ———— 1

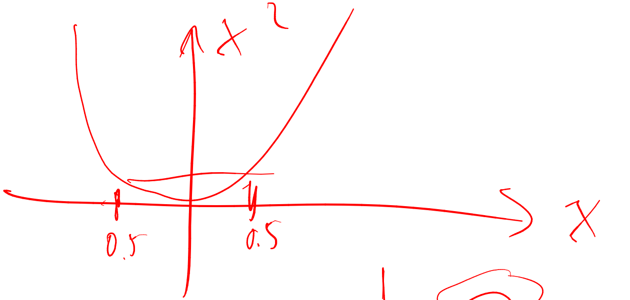
Each individual has a prob. of zero.

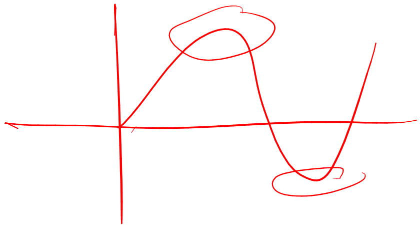
Allowed to say:

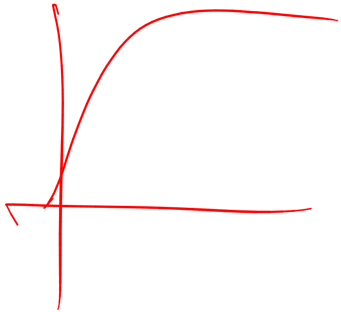
$X \in (a, b)$ has prob. 0.1

$X = 0.125$ has prob ~~0.1~~ 0

Demo: Plotting Distributions with Histograms







Discr.　　　　　　　　Continuous

$p(x)$　　$P(X \in (a,b)) = \int_a^b p(x)\, dx$

Averages: What?

- Define ‘expected value’ of a random variable.

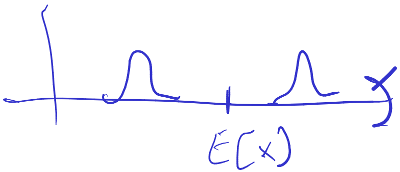
$x_1 \dots x_n$

$p_1 \dots p_n$

$$E[X] = \sum_{i=1}^{N} x_i p_i \qquad\qquad E[X] = \int_{\mathbb{R}} x\, p(x)\, dx$$

- Define variance of a random variable.

$$E[f(x)] = \sum_{i=1}^{N} f(x_i) \cdot p_i \qquad\qquad E[f(x)] = \int f(x) \cdot p(x)\, dx$$



Variance of a random variables

$$E[(X - E(X))^2] = E[X^2] - E(X)^2$$

Normalization

- What is $E[1]$? Yes, the expected value of 1?

$$1 = E[1] = \sum_{i=1}^{n} p_i$$

$$1 = E[1] = \int_{\Omega} p(x)\,dx$$

$$\iint \tilde{p}(x,y)\,dx\,dy = T \cdot \underbrace{E[1]}_{1}$$

$$\tilde{p}(x,y) = \begin{cases} 1 & \text{if } (x,y) \text{ on the DB} \\ 0 & \text{if not} \end{cases}$$

$$\therefore p(x,y) = \frac{1}{T}\tilde{p}(x,y)$$

Expected Value: Example I

- What is the expected snowfall in Champaign?

$$E(\text{Snow}) = \int_{\text{day}} \text{snow}(d) \cdot p(d) \, d\text{day}$$

$$p(d) = \frac{1}{365}$$

Expected Value: Example II

- What is the expected snowfall in Illinois?

$$E[\text{Snow}] = \int\int \text{Snow}(x,y) \cdot p(x,y)\, dx\, dy$$

Tool: Law of Large Numbers

Terminology:

- *Sample*: A random number x_i whose values follow a distribution $p(x)$.

In words:

- As the number of samples $N \to \infty$, the average of samples converges to the expected value with probability 1.

In symbols:

$$P\left[\lim_{N \to \infty} \frac{1}{N}\left(\sum_{n=1}^{N} x_i\right) = E[X]\right] = 1.$$

Or:

$$E[X] \approx \frac{1}{N}\left(\sum_{n=1}^{N} x_i\right)$$

Sampling: Approximating Expected Values

Integrals and sums in expected values are often challenging to evaluate.

- How can we approximate an expected value?

  Idea: Draw random samples. Make sure they are distributed according to $p(x)$.
- What is a Monte Carlo method?

Sampling II: Approximating Expected Values

- What if I *can't* sample from $p(x)$?

  Idea: Draw uniformly distributed random samples.

Demo: Computing π using Sampling
Demo: Errors in Sampling

Sampling III: Importance Sampling

Integrals and sums in expected values are often challenging to evaluate.

Idea: Draw random samples from a sampling distribution q.

1. *Draw N samples* x_i distributed according to $q(x)$.
2. Possibly: Reject sample if $p(x_i) = 0$.
3. Approximate

$$E[f(X)] \approx \sum_{i=1}^{N} f(x_i) \frac{p(x_i)}{q(x)}.$$

- When is this a good way to sample?

Sampling: Error

The Central Limit Theorem states that with

$$S_n := x_1 + x_2 + \cdots + x_n$$

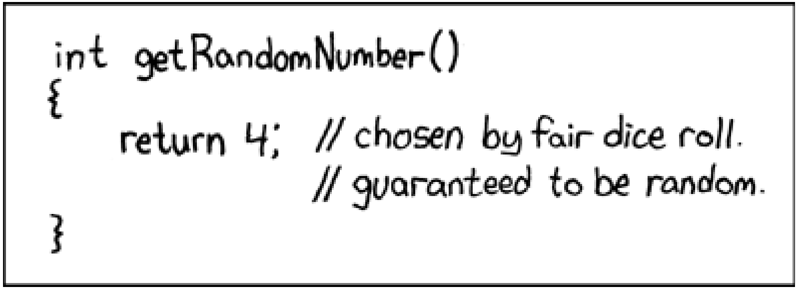
for the (x_i) independent and identically distributed we have that

$$\frac{S_n - n\,E[x_i]}{\sqrt{\sigma^2[x_i]n}} \to \mathcal{N}(0,1),$$

i.e. that term approaches the normal distribution. Or, short and imprecise:

$$\left|\frac{1}{n}S_n - E[x_i]\right| = O\left(\frac{1}{\sqrt{n}}\right).$$

Computers and Random Numbers



[from xkcd]

- How can a computer make random numbers?

Random Numbers: What do we want?

- What properties can ‘random numbers’ have?

What's a Pseudorandom Number?

- Actual randomness seems like a lot of work. How about 'pseudorandom numbers?'

Demo: Playing around with Random Number Generators

Some Pseudorandom Number Generators

Lots of variants of this idea:

- LC: 'Linear congruential' generators
- MT: 'Mersenne twister'

Remarks:

- Initial state and parameter choice often surprisingly tricky.
  Bad choice: Predictable/correlated numbers.
  E.g. Debian OpenSSL RNG disaster
- Absolutely **no reason** to use LC or MT any more. (Although almost all randonumber generators you're likely to find are based on those–Python's `random` module, `numpy.random`, C's `rand()`, C's `rand48()`.
- These are **obsolete.**

Counter-Based Random Number Generation (CBRNG)

- What's a CBRNG?

4 Sources of Error in Computational Models

5 Accuracy and Convergence

6 Floating Point

Wanted: Real Numbers... in a computer

- Computers can represent *integers*, using bits:

$$23 = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = (10111)_2$$

How would we represent fractions?

Fixed-Point Numbers

- Suppose we use units of 64 bits, with 32 bits for exponents $\geqslant 0$ and 32 bits for exponents < 0. What numbers can we represent?

- How many 'digits' of relative accuracy (think relative rounding error) are available for the smallest vs. the largest number?

Floating Point numbers

- Convert $13 = (1101)_2$ into floating point representation.

- What pieces do you need to store an FP number?

In-class activity: Floating Point

Unrepresentable numbers?

- Can you think of a somewhat central number that we cannot represent as

$$x = (1.\_\_\_\_\_\_\_\_\_\_\_)_2 \cdot 2^{-p}?$$

Demo: Picking apart a floating point number

Subnormal Numbers

- What is the smallest representable number in an FP system with 4 stored bits in the significand and an exponent range of $[-7, 7]$?

Running out of digits

- Suppose you store π in a floating point number. What do you get?

Demo: Density of Floating Point Numbers
Demo: Floating Point vs. Program Logic

Floating Point and Rounding Error

What is the relative error produced by working with floating point numbers?

- What is smallest floating point number > 1? Assume 4 bits in the significand.
- What's the smallest FP number > 1024 in that same system?
- Can we give that number a name?
- What does this say about the relative error incurred in floating point calculations?
- What's that same number for double-precision floating point? (52 bits in the significand)

Demo: Floating Point and the Harmonic Series

Implementing Arithmetic

- How is floating point addition implemented?
  Consider adding $a = (1.101)_2 \cdot 2^1$ and $b = (1.001)_2 \cdot 2^{-1}$ in a system with three bits in the significand.

Problems with FP Addition

- What happens if you subtract two numbers of very similar magnitude? As an example, consider $a = (1.1011)_2 \cdot 2^0$ and $b = (1.1010)_2 \cdot 2^0$.

Demo: Catastrophic Cancellation

Part 2:
Arrays–Computing with Many Numbers

7 Modeling the World with Arrays

7.1 The World in a Vector

7.2 What can Matrices Do?

7.3 Graphs

8 Norms and Errors

Recap: Norms

- What's a norm?

- Define norm.

- Examples of norms?

- Does the choice of norm really matter much?

Demo: Vector norms [Make sure this covers unit balls]

Norms and Errors

- If we're computing a vector result, the error is a vector.
  That's not a very useful answer to 'how big is the error'.
  What can we do?

Absolute and Relative Error

- What are the relative and absolute errors in approximating [TODO] the location of Siebel center as ...?

Matrix Norms

- What norms would we apply to matrices?

Demo: Matrix norms

In-class activity: Matrix norms

Properties of Matrix Norms

Matrix norms inherit the vector norm properties:

1. $\|A\| > 0 \Leftrightarrow A \neq \mathbf{0}$.
2. $\|\gamma A\| = |\gamma|\|A\|$ for all scalars γ.
3. Obeys triangle inequality $\|A + B\| \leqslant \|A\| + \|B\|$

- But also some more properties that stem from our definition:

Conditioning

- Now, let's study conditioning of solving a linear system

$$A\boldsymbol{x} = \boldsymbol{b}.$$

Demo: Condition number visualized
In-class activity: Matrix Conditioning
Demo: Conditioning of 2×2 Matrices

Residual and Error

- What is the residual vector of solving the linear system

$$\boldsymbol{b} = A\boldsymbol{x}?$$

- How do the (norms of the) residual vector $\boldsymbol{r}$ and the error $\Delta \boldsymbol{x} = \boldsymbol{x} - \hat{\boldsymbol{x}}$ relate to one another?

9 The ‘Undo’ Button for Linear Operations: LU

10 LU: Applications

10.1 Interpolation

11 Repeating Linear Operations: Eigenvalues and Steady States

12 Eigenvalues: Applications

13 Approximate Undo: SVD and Least Squares

14 Least Squares: Applications

14.1 Data Fitting

Part 3:
Approximation–When the Exact Answer is Out of Reach

15 Iteration and Convergence

16 Solving One Equation

17 Solving Many Equations

18 Finding the Best: Optimization in 1D

19 Optimization in *n* Dimensions