COMPUTING

Statistics	Sampling	Inference
<u>Distributions</u> Measurement Normal Regression <u>Probability</u> Addition Multiplication Variables	<u>Sampling</u> Methods Experiments Distribution <u>Normal Distribution</u> Large Counts Condition Central Limit Theorem	<u>Inference</u> Confidence Interval Significance Test Chi-Square <u>Types</u> Proportions Means Slopes

Language	Computing	Learning
<u>Systems</u> Writing Sounds <u>Structure</u> Morphology Syntax Semantics	<u>Combinations</u> Subsets Permutations Recursion <u>Algorithms</u> Preprocessing Greedy Dynamic	<u>Methods</u> Supervised Unsupervised Reinforcement <u>Algorithms</u> Regression Perceptrons Neural Networks

MEASUREMENT

single		
absolute	$x \pm \Delta x$	absolute error
	$u = 2x$	$\Delta u = 2\,\Delta x$ scaling
relative	$\frac{\Delta x}{x}$	relative error
	$u = 2x \quad \frac{du}{u} = \frac{2dx}{2x} = \frac{dx}{x}$	$\frac{\Delta 2x}{2x} = \frac{\Delta x}{x}$ scaling
	$v = \frac{x}{t} \quad \frac{dv}{v} = \frac{-\frac{x}{t^2}}{\frac{x}{t}} = -\frac{dt}{t}$	$\frac{\Delta(1/x)}{1/x} = \frac{\Delta x}{x}$ reciprocal
	$A = s^2 \quad \frac{dA}{A} = \frac{2sds}{s^2} = 2 \cdot \frac{ds}{s}$	$\frac{\Delta(x^2)}{x^2} = 2 \cdot \frac{\Delta x}{x}$ power

compound		
sum	$a + b$	sum of independent variables
	$\left(\Delta(a+b)\right)^2 = (\Delta a)^2 + (\Delta b)^2$	absolute add in quadrature
	$\Delta(a+b) < \Delta a + \Delta b$	weighted average lowest total
average	$\overline{x} = \frac{x_1 + x_2 + \ldots x_n}{n}$	with same independent errors
	$\Delta\overline{x} = \frac{1}{n} \cdot \left(\sqrt{(\Delta x_1)^2 + \ldots}\right)$	by scaling and sum
	$\Delta\overline{x} = \frac{1}{\sqrt{n}} \cdot \Delta x$	reduced with sample squared
product	$ab = a \cdot b$	product of independent variables
	$\left(\frac{\Delta ab}{ab}\right)^2 = \left(\frac{\Delta a}{a}\right)^2 + \left(\frac{\Delta b}{b}\right)^2$	relatives add in quadrature

DISTRIBUTIONS

single		
statistics	collect and analyze	draw conclusions from data
	categorical and quantitative	tabular graphical numerical
graphical	data center variation	shape symmetric skewed
	patterns clusters gaps	outliers $1.5 \times$ inter quartile
numerical	$\bar{x} = \frac{\sum x_i}{n}$ center	$s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n-1}}$ variation
	$P_k = \frac{(n+1)}{100} k$ percentile	$IQR = Q_3 - Q_1$ inter quartile
	$z\text{-score} = \frac{x - \bar{x}}{s}$	distance in standard deviations
normal	single peak symmetric	magical for statistics
	central limit theorem	sample means always normal
	completely specified (μ, σ)	interval integrals all known
	standard normal $(0, 1)$	$z = \frac{x - \mu}{\sigma}$ conversion

dual		
correlation	numerical relationship	between two variables
	$r = \frac{1}{n-1} \sum \left(\frac{x_i - \bar{x}}{s_x}\right)\left(\frac{y_i - \bar{y}}{s_y}\right)$	
	r^2	proportion accounted for
regression	how one variable changes	in response to other change
	$\hat{y} = a + bx$	predicted value
	$b = r \frac{s_y}{s_x}$	relation to correlation
	$y = a x^b \quad \ln y = \ln a + b \ln x$	transform to linear

PROBABILITY

probability		
probability	random process has	outcomes by chance
	probability is long run	proportion of an outcome
	law of large numbers	proportion approach probability
model	sample space all possible	event subset of outcomes
	$P(A) = \frac{\text{number outcomes in A}}{\text{total outcomes in sample space}}$	

addition		
complement	$P(\overline{A}) = 1 - P(A)$	event does not occur
	$\frac{P(A)}{P(\overline{A})}$	odds in favor
addition	the probability that	either event occurs
	$P(A \cup B) = P(A) + P(B) - P(A \cap B)$	
	$P(A \cap B) = 0$	mutually exclusive

multiplication		
conditional	probability one event happens	given another event happened
	$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$	
independence	whether one event happens	does not affect the other
	$P(A \mid B) = P(A)$	
multiplication	the probability that	both events occur
	$P(A \cap B) = P(A) \cdot P(B \mid A)$	rearrangement
	$P(A \cap B) = P(A) \cdot P(B)$	independent events

VARIABLES

random variable		
discrete	assigns numerical values	to random process outcomes
	probability distribution	possible values and probabilities
	long run outcomes	not individual probabilities
	$\mu_X = E(X) = \sum x_i P(x_i)$	
	$\sigma_X = \sqrt{\sum (x_i - \mu_X)^2 P(x_i)}$	
combinations	combine random variables	linearity of expected value
	addition $X + k \to \mu + k, \sigma$	scaling $kX \to k\mu, k\sigma$
	$aX + bY \to a\mu_X + b\mu_Y, \sqrt{a^2\sigma_X^2 + b^2\sigma_Y^2}$	

repeated		
binomial	fixed number trials	count total successes
	$P(X = x) = \binom{n}{x} p^x (1-p)^{n-x}$	
	$\mu_x = np$	$\sigma_x = \sqrt{np(1-p)}$
geometric	expected number of trials	until first success
	$P(X = x) = (1-p)^{x-1} p$	
	$\mu_x = \frac{1}{p}$	$\sigma_x = \frac{\sqrt{1-p}}{p}$

continuous		
continuous	continous possible values	anywhere on interval
	probability is area	not any instant height
	normal random variable has	normal probability distribution

METHODS

experiment		
data	population is entire group	we want to study
	census data from all	sample data from subset
sampling	use sample survey to	learn about whole population
	random sample	chance process determine sample
experiment	observational study	no attempt influence responses
	experimental study	impost treatments to influence
	observational no causation	experiment distill causation

methods		
SRS	simple random sample	equal chance being chosen
	sample with or without	replacement of chosen
stratified	strata groups of individuals	thought to share characteristics
	combine SRS from each strata	well chosen more precise
cluster	cluster groups of individuals	who are near each other
	more practical to use	for large populations
systematic	ordered population	select every kth individual
	when SRS impossible	such as polling station

bias		
bias	faulty sampling method	likely to mis estimate value
	convenience easy to reach	voluntary those who respond
	undercoverage unlikely chosen	nonresponse unreachable
	response bias	persistent pattern inaccurate

EXPERIMENTS

experiments		
experiments	deliberately setup	comparison groups
	use comparison of results	to infer cause and effect
	impose treatments	on experimental units
	treatment specific condition	single or combo variable values
	experimental unit objects	called subjects if human beings
variables	explanatory variable effect	on response variable value
	confounding if two variables	both associated with third
	factor is explanatory variable	levels are its different values

design		
control	keep environment constant	to distill cause and effect
	confounding remove variation	placebo null treatment
	control group of units	provide baseline for comparison
assignment	treatment is unknown	either single or double blind
	random assignment	treatments assigned randomly
	completely randomized design	no restrictions on pairings
	replication enough units	for statistical significance
	blocks groups of units	thought to be similar
	randomized block design	well chosen more precise
	matched pair design	use blocks of size two

SAMPLING DISTRIBUTION

sampling distribution		
statistics	use data from sample	to conclude about population
	parameter describes	a characteristic of a population
	statistic describes	a characteristic of a sample
sampling distribution	distribution of a statistic	of all possible samples
	mean weight of 100 pennies	all mean weights of 20 pennies
	large populations not possible	approximate by many samples
	distribution is of statistics	not of individual data values

normal distribution		
conditions	sampling distribution	of all possible samples
	is unbiased estimator	and normal distribution if
	sample without replacement	10 % rule $n < 0.1N$
	large counts condition	$np \geq 10$ $n(1-p) \geq 10$
	central limit theorem	$n \geq 30$
proportion	$\mu_{\hat{p}} = p$	$\sigma_{\hat{p}} = \sqrt{\frac{p(1-p)}{n}}$
	$\mu_{\hat{p}_1-\hat{p}_2} = p_1 - p_2$	$\sigma_{\hat{p}_1-\hat{p}_2} = \sqrt{\frac{p_1(1-p_1)}{n_1} + \frac{p_2(1-p_2)}{n_2}}$
mean	$\mu_{\overline{X}} = \mu$	$\sigma_{\overline{X}} = \frac{\sigma}{\sqrt{n}}$
	$\mu_{\overline{X}_1-\overline{X}_2} = \mu_1 - \mu_2$	$\sigma_{\overline{X}_1-\overline{X}_2} = \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}$

CONFIDENCE INTERVAL

inference		
inference	use single sample to	conclude about population
	given sampling distribution	is a normal distribution
	use experiment comparison	infer about cause and effect

confidence interval		
confidence interval	likelihood that interval	contains population parameter
	accept cause and effect claim	if value is inside the interval
	point estimate ± margin of error	
	margin of error = (critical value) (standard error)	
proportion	$\hat{p} \pm z^* \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$	
	$(\hat{p}_1 - \hat{p}_2) \pm z^* \sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n_1} + \frac{\hat{p}_2(1-\hat{p}_2)}{n_2}}$	
mean	$\overline{x} \pm t^* \frac{s_x}{\sqrt{n}}$	$df = n - 1$
	$(\overline{x}_1 - \overline{x}_2) \pm t^* \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$	
slope	$b \pm t^* \frac{s}{s_x\sqrt{n-1}}$	$df = n - 2$

SIGNIFICANCE TEST

significance test		
significance test	assess competing claims	about population parameter
	sample statistics versus null	distance in standard deviations
	$\text{standardized test statistic} = \dfrac{\text{statistic} - \text{parameter}}{\text{standard error of statistic}}$	
	use normalized distance	to look up probability
error	Type I reject null when true	Type II fail reject null when false
	power to avoid Type II	$\text{power} = 1 - P(\text{Type II})$
proportion	$z = \dfrac{\hat{p} - p_0}{\sqrt{\dfrac{p_0(1-p_0)}{n}}}$	
	$z = \dfrac{(\hat{p}_1 - \hat{p}_2) - 0}{\hat{p}_C(1-\hat{p}_C)\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}$	$\hat{p}_C = \dfrac{x_1 + x_2}{n_1 + n_2}$
mean	$t = \dfrac{\bar{x} - \mu_0}{\frac{s_x}{\sqrt{n}}}$	$df = n - 1$
	$t = \dfrac{(\bar{x}_1 - \bar{x}_2) - 0}{s_{\bar{x}_1 - \bar{x}_2}}$	
slope	$t = \dfrac{b - \beta_0}{s_b}$	$df = n - 2$

chi square		
chi square	assess hypothetical distributions	for two categories of data
	random sample 10 % rule	large counts condition all > 5
	$\chi^2 = \sum \dfrac{(O-E)^2}{E}$	$df = (r-1)(c-1)$
	$\text{expected} = \dfrac{\text{(row total)(column total)}}{\text{table total}}$	

WRITING

structure		
structure	Rosetta stone	in order
	chaos and order	random order
categories	pictographic	emulation
	logographic	words
	syllabic	sounds
examples	syllabic Japanese katakana	logo syllabic Chinese
	deaf sign language	blind braille

systems		
alphabetic	sounds	consonant vowel
	Latin	Cyrillic Georgian Armenian
abjad	consonant characters	vowels diacritics
	Semitic	Arabic Hebrew
abugida	consonant base	default vowel
	South East Asia	Cree mirror rotation
featural	phonetic	sound production
	Korean	hangul

SOUNDS

syllables		
consonants	constriction	air obstacle
	articulation	place manner voicing
vowels	no constriction	narrow cavity
	features	backness height roundness
	long vowel	two vowels dipthong
syllable	sound group	pronounced together
	onset	consonant c-
	rhyme	nucleus coda -a-t
syllabification	nucleus vowel	$VV \rightarrow V.V$
	onset consonant	$VCV \rightarrow V.CV$
	cluster onset	$VCCV \rightarrow VC.CV$

versification		
versification	verse	series of lines
	meter	weight pattern
weight	onset	irrelevant
	nucleus	short long
	coda	open closed
schemas	vowel length	light short
	coda existance	light no coda
	rhyme length	light single element
meter	sequence	light heavy
	equivalence	one heavy two lights

NOUNS

nouns		
nouns	noun	object places beings
	proper noun names	common noun classes
noun phrase	head noun	modifiers
	the good **boy**	the **boy** inside the house
sentence	noun phrase	verb phrase
	a boy eats the apples	$NP + VP(V + NP)$

modifiers		
morphemes	smallest	function groups
	stem affixes	un-desire-able-ity
different	same forms	different functions
	different morphemes	cat**s** see**s**
allomorphs	different forms	same function
	same morphemes	**il**legal **im**possible

variables		
variables	noun features	that can change
	different forms	same function
allomorphs	definitive	the a
	numbers	singular plural few
	gender	masculine feminine
	class	humans mammals birds
	case	subjective objective possessive

VERBS

verbs		
verbs	verb	action existence state
	arguments	subject object
verb phrase	main verb	modifiers
	transitive	direct object
sentence	noun phrase	verb phrase
	a boy eats the apples	$NP + VP(V + NP)$

modifiers		
morphemes	smallest	functional groups
	main affixes	a boy eats
different	same forms	different functions
	different morphemes	cat**s** see**s**
allomorphs	different forms	same function
	same morphemes	he eat**s** they eat

variables		
variables	verb features	that can change
	different forms	same function
allomorphs	tense	eat ate will eat
	aspect	have eaten am eating
	mood	can may if must would should
	arguments	person number gender
	semantic	intensifiers dynamic agent

STRUCTURE

syntax		
syntax	word order	subject verb object
	fixed flexible	$SVO, SOV > VSO$
arguments	subject	intrasitive verb subject
	agent	transitive verb subject
	object	transitive verb direct object
alignment	subject object	$S, A \quad O$
	agent	$A \quad S, O$

semantics		
semantics	word combinations	meaning
	body parts	emotions
graph	nodes	edges
	overlap	two graphs

others		
numbers	word order	bundle marking
	overcounting	subtractive
orientation	absolute	common
	relative	topography
kinship	family tree	six systems
	cousins	parallel cross

EFFICIENCY

efficiency		
programming	be correct	solve the right problem
	and be efficient	as the inputs get very large
efficiency	worst case as size grows	mathematical order
	size can grow	in amount or in values
	10^8 operations a second	complete in 2 seconds

order		
binary search	$n \leq 10^{10}$	$O(\log n)$
simple search count sort	$n \leq 10^7$	$O(n)$
merge sort	$n \leq 10^7$	$O(n \log n)$
selection sort	$n \leq 10^4$	$O(n^2)$
triple for loop	$n \leq 550$	$O(n^3)$
subsets	$n \leq 25$	$O(2^n)$
travelling salesman	$n \leq 10$	$O(n!)$

sub array sum		
sub array sum	for all sub array sums	determine the largest
	check all possible	triple for-loop $O(n^3)$
	check sum same time	double for-loop $O(n^2)$
	maximum at each position	single for-loop $O(n)$

DATA

data		
primitives	numbers text boolean	int long long double
collections	array linked list dynamic array	[] list vector
utility	queues sets	stack queue deque priority set
mixed	maps objects	map struct class
graphs	nodes connected by edges	could be directed or weighted
tree	graph with no orphans or cycles	directed and unweighted

syntax		
integer	int a = -3; 32 bit	$2^{31} \approx 10^{9}$ silent overflow
	unsigned int = 3; 32 bit	2^{32} positive only larger than int
	long long a = 3; 64 bit	$2^{63} \approx 10^{18}$ no silent overflow
pairs	pair<int, string> p;	p = make_pair(10, "ten");
	auto [a, b] = p; retrieve	p.first p.second set and get
arrays	legacy from C	vectors have better protections
	int a [10]; []	int a[3] = {}; [0, 0, 0]
	int grid[10][10];	two dimensional
vectors	dynamic array	with changeable size
	vector<int> v;	vector<int> a(3); [0, 0, 0]
	v.push_back(10); add to end	int i = pop_back(); get from end
	vector<vector<int>> v;	two dimensional
	sort(v.begin(), v.end());	unique(v.begin(), v.end());
graphs	edges of each node	array of dynamic arrays
	matrix of all edges	a double array
	all edges	dynamic array of pairs

LOGIC

logic		
logic	#include <bits/stdc++.h>	int main() { return 0; }
	using namespace std;	can avoid writing std::sort
	cin >> a; cout << a;	sync_with_stdio(0); cin.tie(0);
	freopen("cow.in", "r", stdin);	freopen("cow.out", "w", stdout);

	int func(int x, int y) {
	  return x + y;
	}
		void func(int x, int y) {
		  int z = x + y;
		}

references		
references	link two variables	with flow through updates
	int b = a; copies value	int& b = a; reference other
	if referenced variable changes	so does reference variable

swap	void swap(int& a, int & b) {
	  int temp = a;
	  a = b;
	  b = temp;
	}
		permanently alters pass in array

pairs	auto [a, b] = p;	copies values only
	auto& [a, b] = p;	updates the pair variable

loop		
loop	ranged based loop	do not need to use iterators
	for(auto x : a) through array	for(auto x : v) through vector
	for(auto& [key, value] : m)	iterate referenced variables
	for (auto& p : m)	itereate referenced pair objects

COMBINATIONS

combinations		
subsets	each item is included or not	efficiency is $O(2^n)$
	binary representation	101 three items in or not
	check if modulo 2^n is 1	to see if nth item is in or not
	$\lfloor \frac{5}{2} \rfloor$	removes last item largest digit
permutations	total possible orders of all items	efficiency is $O(n!)$
	next_permutation function	next in lexicographical order

	while (next_permutation(p, p.size())
	{
	  // current permutation is stored in vector p
	}	

recursion		
recursion	iteratively use same function	to solve a problem
	base case and recursive case	call stack with iterative work
	can be used to make	subsets and permutations

	vector<int> subset; int n = 3;
	void search (int k) {
	  if (k == n+ 1) { // process subset }
	  else {
	    // recurse with k in subset
	    subset.push_back(k);
	    search(k+1);
	    // recurse without k in subset
	    subset.pop_back(k);
	    search(k+1);
	  }
	}
	search(1);	

quicksort	divide and conquer	for a more efficient sort
	recursion can be used	for average $O(n \log n)$

ALGORITHMS

algorithms		
preprocessing	reorganize data strategically	query new data to solve
	find magic cutoff of validity	using efficient binary search
	fair rides and server times	aops bronze binary search
	build cumulative $O(n)$	assess any range $O(1)$
	model all solutions	sequentially with extreme upper
	find boundary value	using binary search $O(\log n)$
greedy	repeatedly select local optimal	move on and never look back
	are only approximations	efficient may not be correct
	classes earliest ending	efficient and correct
	knapsack most expensive	efficient not correct
	interval points	sort by edge then greedy
dynamic	buildup solution	from subproblems
	efficient and correct	unlike simulation or greedy
	knapsack store sub results	iterate to build larger results
	keep track of optimal solutions	can be one or two dimensional

others		
geometry	location of all cows grazing	intersection of rectangles
	brute force paint	one fell swoop extreme sides
	gradual by recursive	overlap of rectangle pairs
	location of any cows grazing	union of rectangles
	principle of inclusion exclusion	every subset overlap $O(2^n)$
graphs	shortest path	breadth first search
	complete traversal	depth first search

PROBLEMS

simulation		
weird algorithm cses	step-by-step math	follow steps
shell game	swap shell locations	swap
cow signal	amplify chars	quad for-loop
speeding ticket	sum speed-over	limit and speed arrays
the lost cow	zigzag	sequential moves
bovine shuffle	three shufflings	original ordering
bucket list	buckets needed	buckets needed at all times
measuring traffic	traffic ranges	update each constraint
circular barn	fill circular barn	double for-loop sum
block game	character cards	double for-loop
tic-tac-toe	alphabet tic-tac-toe	check all lines
mowing the field	re-cutting after re-grows	each cell track last cut time
censoring	remove words repeatedly	create result letter by letter
milk measurement	changes to top cow	sweep all changes

complete search		
milk pails	pour without spilling	all combinations
diamond collector	max diamond displayed	all combinations
daisy chains	average petal daisies	triple for-loop
counting liars	check all positions	double for-loop
cow gynmastics	consistent pairs	double for-loop
bovine genomics	locations of spotiness	count non-overlaps
air conditioning II (recursion)	minimum adjustments	recursion, check validity
quality photo aops	gcd neighbour scores	remove primes still optimal

complete search		
cheese block usaco	completed	completed
it's mooing time usaco	completed	completed
it's mooing time II usaco	completed	completed

preprocessing		
sweep line	entry and exit times	iterate and tabulate
scheduling	ending times	greedy algorithm
tasks and deadlines	durations	increase if swap with shorter
stuck in a rut	stop eating if eaten	sort by collisions
standup applause aops	standing ovation shyness	sort by shyness
cross road III	queue times	pair, sort, sum times
cow college	maximize tuition	sort, single for-loop
where am i	sets & maps	completed
horses fair rides aops	horses qualify for rides	cumulative frequency cutoff
which server aops	who serves Bessie	total time cutoff

greedy		
drink dispenser aops	edge constraints	fill up each step
mooloo	completed	completed
astral superposition usaco	completed	completed

ARTIFICIAL INTELLIGENCE

ai		
ai	symbolic formal logic	deterministic rules
	subsymbolic unconscious	probabilistic distributions
probabilistic	perceptrons	inputs to output
	neural networks	convolution layers
	language model	memory for context
ml	ai decision making	mimic human processes
	ml based on data	remember formulate predict
	models learn then predict	labeled or unlabeled data

learning		
supervised	regression for numeric	linear regression tree ensemble
	classification for categoric	network tree vector ensemble
unsupervised	unlabeled data	cluster reduce generate
	also preprocess data	more effecctive supervised
reinforcement	no data	environmental response
	dynamic update states	exploitation vs exploration

regression		
regression	closer to random point	iterate update intercept slope
	absolute fixed increments	squared proportional error
selection	choose optimal model	performance and complexity
	performance	regression validation error
	complexity	absolute square coefficients

ALGORITHMS

perceptrons		
perceptrons		