Computer Representations

Error ← This is an aside

When approximating things, we want to know how close our answer is to being correct. We use <u>error</u> to quantify that.

<u>Absolute error</u> is the absolute value of the difference b/w the <u>approximated value</u> $\tilde{x}$ & <u>actual value</u> x.

$|\tilde{x} - x|$

$\tilde{x}$ ← indicated w/ a tilde

<u>Relative error</u> is

$$\frac{|\tilde{x} - x|}{|x|}$$

For relative error to be meaningful, we need a meaningful zero. So for example Farenheit/Celcius doesn't have a meaningful zero.

Absolute error has units while relative error is unitless.

Integer Representation

Our goals for integer representations in computers are to only use 1/0 bits, one bitstring represents 1 value, comparisons should be cheap, & all bitstrings should have the same length.

The gold standard for this is 2's complements representation w/ powers of two bitstring lengths.

Real Number Representation

Our goals for representing reals are to store extremely large & extremely small numbers relatively accurately, be a fixed length, & cheap to do arithmetic w/.

The gold standard for this is IEEE 754 floating point numbers. They aren't perfect but they do a great job & are hugely popular.

There are 3 main binary formats
- binary 32 / single precision
- binary 64 / double precision
- binary 128 / quadruple precison

IEEE 754 numbers are split into 3 parts: the (s)ign bit, (e)xponent bits, (f)ractional bits. ← aka significant or mantissa.

Precision	# bits	# bits per part s	e	f	bias (c)
single	32	1	8	23	127
double	64	1	11	52	1023
quadruple	128	1	15	112	16383
MA402 Tiny	6	1	3	2	3

← Bespoke teaching/testing format

To represent both positive & negative exponents we use a bias (rather than 2's complement).

To determine the value of a number x you do

$$x = (-1)^s (1.f)_2 \, 2^{e-c}$$

(concatenate bitstring)

Note:
Floating points are Not reals.
- Floats are all rational (except $\pm\infty$, NaN).
- $\mathbb{R}$ is uncountably infinite & floats are finite.

Floating Point Oddities / Special Numbers

There is a positive & negative zero. Since the default scheme doesn't allow for 0, if you make the exponent & fraction all 0s, you get ± 0 (based on the sign bit).

There is a positive & negative infinity if your number is out of range & in some arithmetic (below). Its exponent & fraction are all 1s.
- $+1/+0 = +\infty$
- $-1/+0 = -\infty$
- $+1/-0 = -\infty$
- $-1/-0 = +\infty$

There is not a number / NaN which is where the exponent is all 1s. The sign & fraction can be anything. All operations on NaN return NaN. It is the catch-all error number.

De-normalized numbers (or subnormal) are a way to extend the range of floats at the expense of losing precision on the low end. When $e=0$ & $f \neq 0$, the leading 1 is dropped & the exponent is the 1-bias. (They are smaller)

The realmax is the largest non-infinite floating point numbers

The realmin is the smallest normalized number.

We define the epsilon of a float x are the distance from x to the next larger number.

$\text{eps}(x) = \varepsilon$

This is also called machine precision & is always a power of two that double when the mantissa overflows.

eps(0) is the smallest floating point number (including denormalized ones)

flintmax is the largest consecutive integer. (For double precision it is $2^{53} \approx 9.0 \cdot 10^{15}$)

If a nu

Rounding

What do we do when we land b/w 2 floating point numbers after doing arithmetic?

First you round to the closest, unless there is a tie. When there's a tie the "even" one wins. That is you round to the number that has a 0 as the last bit. This makes sense b/c IEEE 754 floats alternate the last bit b/w 0 & 1. This means all powers of 2 are "even".

$1 + \varepsilon/2 = 1$ $\quad (1+\varepsilon) + \varepsilon/2 = 1 + 2\varepsilon$

Def:

$u = \varepsilon/2$ is called the unit roundoff, essentially the relative precision of floating point numbers.

For any $x \in \mathbb{R}$ 011001

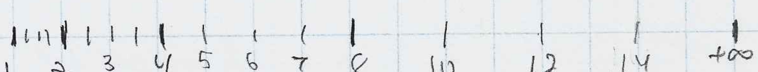
$$\frac{|\text{float}(x) - x|}{|x|} \leq u$$

This is why we know double precision is accurate to ~16 digits.

IEEE 754 numbers follow the rule that for op $\in \{+, -, \times, /\}$ float(x op y) is the true value of (x op y) & then rounded correctly. Same for sqrt.

Practice Problems:

Here we'll use MA 402 Tiny floats Round Down in tie



a) $7 - 1.25 = 6$
b) $(1+7) + 2 = 10$
c) $1 + (7+2) = 8$ (round down!)
d) $(1.25 \cdot 3) \cdot 3 = 4 \cdot 3 = 12$
e) $1.25 \cdot (3 \cdot 3) = 1.25 \cdot 8 = 10$

Note that $+$ & $\cdot$ are not (formally) associative on floats!

Summary

Here's some good things IEEE 754 does:

1) Barring overflow/underflow: $\sqrt{x^2} = |x|$ & $\frac{|x|}{\sqrt{x^2+y^2}} \leq 1$

2) No matter what, $x = y \Leftrightarrow x - y = 0$

This wouldn't be possible w/o subnormal numbers

3) Multiplication/division by powers of 2 is exact (b/c only e changes)

Here's some bad things IEEE 754 does:

1) If $|x \pm y| \ll \min(|x|, |y|)$ we lose most of our sig figs. In other words, numbers of vastly different magnitudes being added loses relative precision. Catestrophic Cancellation

Precision & Functions on Floats

If $\tilde{x}$ is close to x, how close is $\tilde{y} = f(\tilde{x})$ to $y = f(x)$?

Given approximation f^* for f, how close is $f^*(x)$ to $f(x)$?

Def:

If we can find a $\kappa_{abs} \geq 0$ st $|y - \tilde{y}| \leq \kappa_{abs} |x - \tilde{x}|$ the absolute condition number of f at x.

If instead we have $\kappa_{rel} \geq 0$ st $\frac{|y - \tilde{y}|}{|y|} \leq \kappa_{rel} \frac{|x - \tilde{x}|}{|x|}$ the relative condition number of f at x.

We generally want a low κ so that we keep most precision.

Example:

If $x \approx 2$, what is $\sqrt{x}$? About 1.4 κ is small

If $x \approx 10^{10}$, what is $\cos(x)$? Depends on how close x is to 10^{10}? If we have $x \pm 4$ then we don't know $\cos(x)$. κ is large

If $x \approx 3$, what is $f(x)$ where $f(x) = \begin{cases} 1 & \text{if } x \text{ rational} \\ 0 & \text{if } x \text{ is irrational} \end{cases}$? $f(x)$ is pathological & ill-conditioned. $\kappa = \infty$?

Def:

Formally, the relative condition number of f at x is $\lim_{\varepsilon \to 0} \sup_{|\delta x| \leq \varepsilon} \frac{\frac{|\delta y|}{|y|}}{\frac{|\delta x|}{|x|}}$ where $\tilde{x} = x + \delta x$ & $\tilde{y} = y + \delta y$.

That is how much at most is the input error magnified by f.

For differentiable functions we use linearization to find the condition numbers.

$$|f(\tilde{x}) - f(x)| \approx |f'(x)|\,|x - \tilde{x}|$$

κ_{abs} or absolute condition number

$$\frac{|f(\tilde{x}) - f(x)|}{|f(x)|} = \frac{|x f'(x)|}{|f(x)|}\,|x - \tilde{x}|$$

κ_{rel} or relative condition

That is

$$\boxed{\kappa_{abs} = |f'(x)| \quad \& \quad \kappa_{rel} = \frac{|x f'(x)|}{|f(x)|}}$$

Examples:

$f(x) = \alpha x$

$\kappa_{abs} = |f'(x)| = \alpha$ & $\kappa_{rel} = \frac{|x f'(x)|}{|f(x)|} = \frac{|\alpha x|}{|\alpha x|} = 1$

$f(x) = \sqrt{x}$

$\kappa_{abs} = |f'(x)| = \frac{1}{2\sqrt{x}}$ & $\kappa_{rel} = \frac{|x f'(x)|}{|f(x)|} = \frac{|x \frac{1}{2\sqrt{x}}|}{|\sqrt{x}|} = \frac{1}{2}$

$f(x) = \cos(x)$

$\kappa_{abs} = |f'(x)| = |\sin(x)| \leq 1$ & $\kappa_{rel} = \frac{|x f'(x)|}{|f(x)|} = \left|x \frac{\sin(x)}{\cos(x)}\right| = |x \tan(x)|$

Example: Catastrophic Cancellation

Consider $f(x) = x \pm y$.

$\tilde{x} = x(1+\delta_1)$ & $\tilde{y} = y(1+\delta_2)$ where δ_1 & δ_2 are relative errors in x & y.

Recall the triangle inequality: $|x \pm y| \leq |x| + |y|$.

Let's find the relative error

$$\frac{|f(\tilde{x}, \tilde{y}) - f(x,y)|}{|f(x,y)|} = \frac{|x\delta_1 \pm y\delta_2|}{|x \pm y|} \leq \frac{|x\delta_1| + |y\delta_2|}{|x \pm y|} \leq \underbrace{\max(|\delta_1|, |\delta_2|)}_{\text{relative error in input}} \cdot \frac{|x| + |y|}{|x \pm y|}$$

In general, be very careful about cancellation

We have two scenarios: $|x \pm y| = |x| + |y| \Leftrightarrow \kappa = 1$:) in inputs

$|x \pm y| < |x| + |y| \Leftrightarrow \kappa \geq 1$:(

If $|x \pm y| \ll |x| + |y|$, then $\kappa \gg 1$. That is we may lose most precision

Functions vs Algorithms

Functions are purely mathematical. In this class we view them as a binary relation b/w sets.

Algorithms on the other hand are concrete & computational. They are a set of (computer) instructions to execute. These are denoted as functions w/ stars *.

Example:
Let's write a few algorithms for $f(x) = x$.

$f_1^*(x) = x$

$f_2^*(x) = (10^{10}+1) * x - 10^{10} * x$

$f_3^*(x) =$
```
for i = 1:52
    x = sqrt(x)
end
for i = 1:52
    x = x^2
end
x
```

f_1^* is perfect. f_2^* is poor. f_3^* is horrendous.

Def:
A function is well-conditioned iff κ is small.

Def:
An algorithm f^* is forward stable iff

$$\frac{|f^*(x) - f(x)|}{|f(x)|} \leq \underbrace{c}_{c \leq 10^3} \kappa$$

If f is ill-conditioned, you're screwed for all algorithms.

If f is well-conditioned but your algorithm f^* is unstable, you might be able to find a better algorithm.

Example:
Find algorithm for $f(x) = 1 - \cos(x)$.

$f^*(x) = \frac{\sin^2(x)}{1+\cos(x)}$ is much more stable than $f^* = 1 - \cos(x)$ b/c it is more stable.

Note:

$$1 - \cos(x) = 1 - \cos(x) \cdot \frac{1+\cos(x)}{1+\cos(x)} = \frac{1 - \cos^2(x)}{1+\cos(x)} = \frac{\sin^2(x)}{1+\cos(x)}$$

Example:
Recall our example early in class for approximating π.
We did this but had horrible precision. Lets see why.

$$s_{n+1} = \sqrt{2 - \sqrt{4 - s_n^2}}$$

$\sqrt{4-s_n^2}$: fine, $s_n^2 \ll 4$

$2 - \sqrt{4-s_n^2}$: terrible! $\sqrt{4-s_n^2} \approx 2$

Here you'll notice that we have catastrophic cancellation w/ $2 - \sqrt{4-s_n^2}$. We can fix this w/ algebraic manipulation.

$$s_{n+1}^2 = 2 - \sqrt{4-s_n^2}\left(\frac{2+\sqrt{4-s_n^2}}{2+\sqrt{4-s_n^2}}\right) = \frac{4-4+s_n^2}{2+\sqrt{4-s_n^2}} = \frac{s_n^2}{2+\sqrt{4-s_n^2}}$$

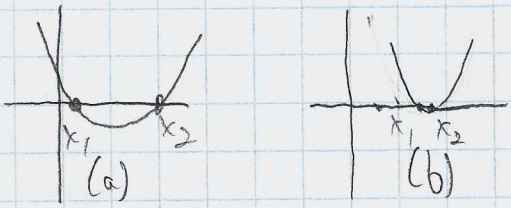
If we code this up you'll notice we can get all 16 digits of π we can get for double precision.

In general, large almost equivalent numbers cancelling out is bad.

Example: Quadratic Equations
If you naively code the quadratic equation as

$$r = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$$

you will get noticeable error in two cases when one of the roots is close to zero (a) & when the roots are near (b).



In case (a) we can get around this by avoiding cancellation & in case (b) the function is just ill conditioned so we're sol.

In case (a), do

$$x_1 = \frac{-b - \text{sign}(b)\sqrt{b^2-4ac}}{2a}$$

This has very little cancellation. Then we know

$$x_2 = \frac{c}{a x_1}$$

Example:

Sometimes you can't get an algorithm to work well in all cases.

Consider $f(x) = x - \sqrt{x^2-1}$.

When $x<<0$ this is okay b/c you don't have any cancellation.
When $x>>0$ this is bad b/c you have cancellation w/ $x - \sqrt{x^2-1}$ b/c $\sqrt{x^2-1} \approx x$.

We can fix case 2 by multiplying by the conjugate

$$f(x) = x - \sqrt{x^2-1}\left(\frac{x+\sqrt{x^2-1}}{x+\sqrt{x^2-1}}\right) = \frac{x^2 - x^2 + 1}{x+\sqrt{x^2-1}} = \frac{1}{x+\sqrt{x^2-1}}$$

However, here $x<<0$ is bad but $x>>0$ is good.

If you were to implement this switching b/w the two implementations based on the sign would be good.

Note: When $|x| >$ flintmax this is bad for both $\pm\infty$.

Norms & Vectors

We want to generalize the idea of magnitudes/lengths of vectors. We do this w/ norms.

Def:
A norm is a function $f: \mathbb{R}^n \to \mathbb{R}$ w/ the following properties for all $u, v \in \mathbb{R}^n$ & $a \in \mathbb{R}$.

- Positive: $f(v) \geq 0$
- Positive Definite: $f(v) = 0 \Leftrightarrow v = \vec{0}$
- Absolute Scaling: $f(a \cdot v) = |a| f(v)$
- Triangle Inequality: $f(u+v) \leq f(u) + f(v)$

In general norms are written as $\|\cdot\|$.

Let's generalize errors & distances w/ norms as they worked w/ $\mathbb{R}$ & $|\cdot|$.

$\frac{\|\tilde{x} - x\|}{\|x\|}$ = relative error of $\tilde{x}$ wrt $\|\cdot\|$.

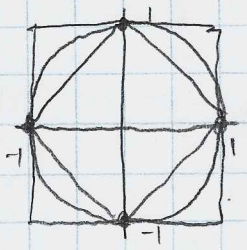
$d(x, y) = \|x - y\|$

Thm:
For 2 norms $\|\cdot\|_1$ & $\|\cdot\|_2$ on $\mathbb{R}^n$, there exist constants $C, D \in \mathbb{R}$ so that for all $x \in \mathbb{R}^n$

$$C\|x\|_1 \leq \|x\| \leq D\|x\|_2$$

Def:
Given a norm $\|\cdot\|$, the unit ball $\|\cdot\|$ is a set of points $P = \{x \in \mathbb{R}^n \mid \|x\| = 1\}$ where the norm is 1.

Here is the unit ball of 3 different norms



Def: Special Ns
We have a few special, named norms.

- $\|x\|_2$ = Euclidean Norm (sqrt of sum of squares, normal)
- $\|x\|_1$ = Manhattan Norm (sum of absolute values)
- $\|x\|_\infty$ = Infinity Norm (max of absolute value

Rem:

$\|\cdot\|_2$ is special b/c it comes from the dot product

$\|x\|_2 = \sqrt{x \cdot x} = \sqrt{x^T x} \quad \forall x \in \mathbb{R}^n$

This gives us the Cauchy-Schwartz inequality $\forall x, y \in \mathbb{R}^n$

$|x \cdot y| = |x^T y| \leq \|x\|_2 \|y\|_2$

& also

$x \cdot y = x^T y = \|x\|_2 \|y\|_2 \cos(\theta)$ where θ is the angle b/c x & y

$\|\cdot\|_2$ also has the Pythagorean theorem. That is $\forall x, y \in \mathbb{R}^n$ where $x^T y = 0 \Leftrightarrow x \perp y$

$\|x + y\|_2^2 = \|x\|_2^2 + \|y\|_2^2$

Example:

Let's try to construct our own norm.

Let $f(x) = x^2$. Is f a norm?

We can easily show f satisfies the positive & positive definite properties.

However, f fails both the absolute scaling & triangle inequality.

$f(2 \cdot 1) = 2^2 = 4$

$|2| f(1) = |2| \cdot 1^2 = 2$

$4 \neq 2$ so f fails absolute scaling

$f(1+1) = 2^2 = 4$

$f(1) + f(1) = 1^2 + 1^2 = 2$

$4 \neq 2$ so f fails triangle inequality.

Example:

Consider $f\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 2|x_1| + 3|x_1 - x_2|$. Is f a norm?

First, f is clearly positive.

Let's show f is positive definite. Clearly $f\begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0$, so let's show $f(x) = 0 \Rightarrow x = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

$f\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 2|x_1| + 3|x_1 - x_2| = 0$

$\Rightarrow |x_1| = 0 \Rightarrow x_1 = 0$

$|x_1 - x_2| = 0 \Rightarrow |-x_2| = 0 \Rightarrow x_2 = 0$

So f satisfies the positive definite property.

Now we show absolute scaling. Consider $x \in \mathbb{R}^2$ & $\alpha \in \mathbb{R}$

$f(\alpha x) = f\left(\alpha \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = f\begin{bmatrix} \alpha x_1 \\ \alpha x_2 \end{bmatrix} = 2|\alpha x_1| + 3|\alpha x_1 - \alpha x_2|$

$= 2|\alpha||x_1| + 3|\alpha||x_1 - x_2|$

$= |\alpha|(2|x_1| + 3|x_1 - x_2|)$

$= |\alpha| f(x)$

Now we show the triangle inequality (on the next page).

Consider some $x, y \in \mathbb{R}^2$

$$\begin{aligned} f(x+y) &= 2|x_1 + y_1| + 3|x_1 + y_1 - x_2 - y_2| \\ &= 2|x_1 + y_1| + 3|(x_1 - x_2) + (y_1 - y_2)| \\ &\leq 2|x_1| + 2|y_1| + 3|x_1 - x_2| + 3|y_1 - y_2| \\ &= 2|x_1| + 3|x_1 - x_2| + 2|y_1| + 3|y_1 - y_2| \\ &= f(x) + f(y) \end{aligned}$$

Thus f satisfies the triangle inequality.

Since f satisfies all necessary properties, f is a norm.

Intro to Probability

A <u>sample space</u> Ω is the set of possible outcomes of an experiment.
An <u>event</u> E is a set of zero or more outcomes.
The <u>probability</u> of an event $P(E)$ describes how likely an event is. Formally, it is a function $P: \text{Pow}(\Omega) \to \mathbb{R}$ where

- $0 \le P(E) \le 1 \quad \forall E \subseteq \Omega$
- $P(E_1 \sqcup \dots \sqcup E_n) = P(E_1) + \dots + P(E_n)$ Note: $E_i \cap E_j = \varnothing \quad \forall i,j = 1..n$
- $P(E) = 1 \Leftrightarrow E = \Omega$ & $P(E) = 0 \Leftrightarrow E = \varnothing$

<u>Thm:</u>
For any event E in sample space Ω.
$$P(E^c) = 1 - P(E)$$

<u>Pf:</u>
$E \cap E^c = \varnothing$ so
$$P(E) + P(E^c) = P[E \sqcup E^c] = P(\Omega) = 1$$
Thus
$$P(E^c) = 1 - P(E)$$

<u>Thm:</u>
This is the inclusion-exclusion theorem.

For any two events E_1 & E_2 in sample space Ω
$$P(E_1 \cup E_2) = P(E_1) + P(E_2) - P(E_1 \cap E_2).$$

<u>Thm:</u>
Suppose sample space Ω is finite & each outcome is equally likely, then for any event $E \subseteq \Omega$
$$P(E) = \frac{|E|}{|S|}$$

<u>Def:</u>
Consider some events A & B in sample space Ω are <u>independent</u> iff
$$P(A \cap B) = P(A)\,P(B)$$

For intuition, this means that A happening gives no information about B happening & vice versa.

Def:
A random variable (RV) is a function $X: \Omega \to \mathbb{R}$ that takes outcomes in the sample space & returns some IR count/property.

X is discrete iff it takes on countably many distinct values (including finite) & continuous otherwise.

Def:
A probability mass function (PMF) $P_X: \mathbb{R} \to [0,1]$ is a function that takes in a RV's output & returns its probability. (discrete RVs)

Example:
Consider a game where you roll a die & get -1$ if you roll a 1, 2, or 3, 1$ if you roll a 4 or 5, & 3$ if you roll a 6.

The RV is discrete here & is given as

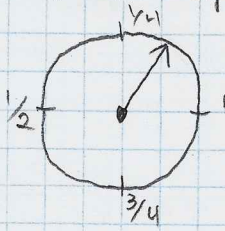
$$X(e) = \begin{cases} -1 & \text{if } e = 1,2,3 \\ 1 & \text{if } e = 4,5 \\ 3 & \text{if } e = 6 \end{cases}$$

The PMF P_X for this RV is

$$P_X(a) = \begin{cases} 1/2 & \text{if } a = -1 \\ 1/3 & \text{if } a = 1 \\ 1/6 & \text{if } a = 3 \end{cases}$$

We can extend this concept easily to continuous RVs.

Example:
Consider spinning a spinner. Let X be where the spinner lands.



Note $X \sim \text{uniform}(0,1)$ is a uniform distribution

Let's find some probabilities on X

$P(0 \le X \le 1) = 1$
$P(X = 0) = 0$
$P(X = 1/2) = 0$
...

Notice that every individual point has probability 0 but areas don't. How do we reconcile this? Calculus!

Def:
A Cumulative density function (CDF) $F_X: \mathbb{R} \to [0,1]$ is a function that takes in the output of a continuous random variable & returns the probability that a random element is smaller than the given.
$F_X(a) = P(X \le a) = P(X < a)$.

Def:
A probability density function (PDF) is the derivative of a CDF

Remark:

$\forall\ a \le b$ & RVs X

i) $P(a < X \le b) = F_X(b) - F_X(a)$

ii) $f_X(a) = \frac{d}{da} F_X(a)$ if the derivative exists

iii) $F_X(a) = \int_{-\infty}^{x} f_X(t)\,dt$ if f_X exists

iv) F_X is monotonically increasing (but not strictly increasing)

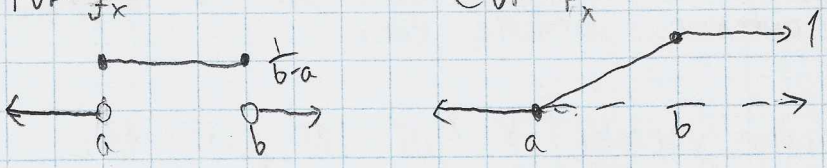
v) $\lim_{a \to -\infty} F_X(a) = 0$

vi) $\lim_{a \to \infty} F_X(a) = 1$

Common Continuous Distributions

Uniform Distribution $X \sim U(a, b)$

PDF f_X CDF F_X



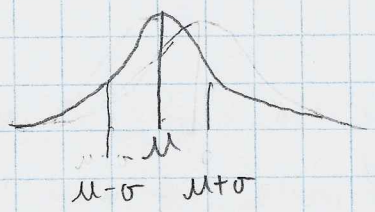
Normal / Gaussian Distribution $X \sim N(\mu, \sigma^2)$.

We call μ the mean & σ^2 the variance.

$$f_X(t) = \frac{1}{\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \leftarrow \text{PDF}$$

We call $N(0,1)$ the **standard normal distribution**.

PDF f_X



Example:

Consider a dartboard of radius $r=1$. Let R be the distance of a dart from the center (where the darts land uniformly & randomly).

The sample space Ω is

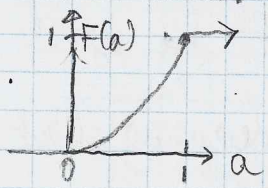
$$\Omega = \{(x, y) \mid x^2 + y^2 \le 1\} = \{\vec{x} \in \mathbb{R}^2 \mid \|\vec{x}\| \le 1\}$$

B/c the darts land uniformly, for $E \subseteq \Omega$.

$$P(E) = \frac{\text{area}(E)}{\text{area}(\Omega)}$$

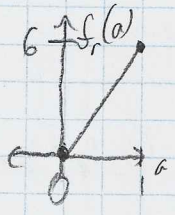
We now find the CDF $F_R(a)$, which is the probability the dart lands w/in a of the dartboard's center.

$$F_R(a) = P(R \le a) = \frac{\pi a^2}{\pi 1^2} = a^2$$



Now we find the PDF $f_R(a)$.

$$f_R(a) = F_R'(a) = \begin{cases} 0 & a < 0 \\ 2a & 0 \le a \le 1 \\ 0 & a > 1 \end{cases}$$



Summary Statistics

The idea w/ summary statistics is to give you an idea about the experiment. So for example the expected outcome, variance, etc.

Def:

Given some discrete random variable X w/ PMF $p_X(t)$, the expected value $E[X]$ & for $E[g(X)]$ where g is a function is

$$E[X] = \sum t\, p_X(t) \qquad E[g(X)] = \sum g(t)\, p_X(t)$$

Similarly for continuous RV X w/ PDF $f_X(t)$, $E[X]$ & $E[g(X)]$ is

$$E[X] = \int_{-\infty}^{\infty} t\, f_X(t)\, dt \qquad E[g(X)] = \int_{-\infty}^{\infty} g(t)\, f_X(t)\, dt$$

Thm:

The expected value is linear, so $\forall \alpha, \beta \in \mathbb{R}$ & RVs X, Y

$$E[\alpha X + \beta Y] = \alpha E[X] + \beta E[Y]$$

By extension for coefficients $\alpha_1, \dots, \alpha_n \in \mathbb{R}$ & RVs $X_1, \dots, X_n$

$$E\left[\sum_{i=1}^{n} \alpha_i X_i\right] = \sum_{i=1}^{n} \alpha_i E[X_i]$$

Physically, you can view the expected value as the center of mass

Now that we have the idea of center/expected value, we want to get the idea of spread/distance from the expected value.

Def:

The mean absolute deviation is a rare way of measuring deviation.

Let X be a RV & $\mu = E[X]$.

$$D[X] = E[|X - \mu|].$$

Def:

The common way to do this is variance. Let X be a RV & $\mu = E[X]$

$$\sigma^2 = \text{Var}[X] = E[(X-\mu)^2]$$

We define standard deviation as: $\sigma[X] = \sqrt{\text{Var}[X]}$

Note that variance is much more sensitive to extremes. Here's a table of similarities

Measure	Physical Analogy	Similar Norms
$E[X]$	center of mass	
$D[X]$		$\|\|x\|\|_1$
$Var[X]$	moment of inertia	$\|\|x\|\|_2^2$
$\sigma[X]$		$\|\|x\|\|_2$

Remark:
Let X be a RV & $\alpha, \beta \in \mathbb{R}$
$\sigma[\alpha X + \beta] = |\alpha| \sigma[X]$ ← absolute scaling
$Var[\alpha X + \beta] = \alpha^2 Var[X]$

Note: Normal/Gaussian distributions are closed under shifting/scaling.

Def:
Two RVs X & Y are independent iff
$P(X \le a \ \& \ Y \le b) = P(X \le a) \cdot P(Y \le b) \quad \forall a \in \Omega_X \ \& \ b \in \Omega_Y$
or equivalently w/ CDFs F_X & F_Y & $F_{X,Y}$
$F_{X,Y}(a, b) = F_X(a) F_Y(b) \quad \forall x \in \Omega_X \ \& \ y \in \Omega_Y$ (already covered this)

Intuitively X gives no info about Y & vice versa.

Thm:
Given 2 independent RVs X, Y, then
i) $E[XY] = E[X] \cdot E[Y]$
ii) $V[X+Y] = V[X] + V[Y]$ ← Pythagorean Theorem

Def:
$X_1, \dots, X_n$ are <u>mutually independent</u> if knowing $(n-1)$ of the RVs outcomes "gives no information" about the last one.

Note that is not sufficient or necessary that the RVs be pairwise independent.

Thm:
If $X_1, \dots, X_n$ are pairwise independent RVs, then
$V[X_1 + \dots + X_n] = V[X_1] + \dots + V[X_n]$.

Def:
We say $X_1, \dots, X_n$ are independent, identically distributed (iid) iff
i) They are mutually independent.
ii) They all have the same CDF.

Example:
Roll a die 100 times (or 100 dice). The rolls' outcomes $X_1, \dots, X_{100}$ are iid.

Law of Large Numbers

Thm: Weak Law of Large Numbers
Let $X_1, \dots, X_n$ be iid random variables w/ distribution X.
Let

$$\bar{X}_n = \frac{X_1 + \dots + X_n}{n}$$

Then $\forall \varepsilon > 0$

$$\lim_{n \to \infty} (|\bar{X}_n - \mu| > \varepsilon) = 0$$

Informally, as we trend towards infinity, we approach μ perfectly (on average).

Recall the normal/Gaussian distribution $N(\mu, \sigma^2)$ is described by the PDF

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

Thm: Central Limit Theorem (CLT) (don't have to be normal)
Let $X_1, \dots, X_n$ be a sequence of iid random variables (RVs) w/ distribution X.
Let $\mu = E[X]$ & $\sigma^2 = \mathrm{Var}[X]$.
Let

$$\bar{X}_n = \frac{X_1 + \dots + X_n}{n} \quad \& \quad Z_n = \frac{\sqrt{n}}{\sigma}(\bar{X}_n - \mu)$$

As $n \to \infty$, $Z_n \to N(0,1)$ or equivalently $\bar{X}_n \to N(\mu, \sigma^2)$

Pf:

$$E[\bar{X}_n] = \frac{E[X_1] + \dots + E[X_n]}{n} = \boxed{\mu = E[\bar{X}_n]}$$

$$\mathrm{Var}[\bar{X}_n] = \mathrm{Var}[\tfrac{1}{n}(X_1 + \dots + X_n)] = \tfrac{1}{n^2}\mathrm{Var}[X_1 + \dots + X_n] = \tfrac{1}{n^2}(\mathrm{Var}[X_1] + \dots + \mathrm{Var}[X_n])$$

(bc independent)

$$= \frac{1}{n^2}(n\sigma^2)$$

$$\boxed{\mathrm{Var}[\bar{X}_n] = \frac{\sigma^2}{n}} \quad \& \text{ so } \quad \boxed{\mathrm{Stddev}[\bar{X}_n] = \frac{\sigma}{\sqrt{n}}}$$

$$E[Z_n] = \frac{\sqrt{n}}{\sigma}(E[\bar{X}_n] - \mu) = 0$$

$$\mathrm{Var}[Z_n] = \mathrm{Var}\left[\frac{\sqrt{n}}{\sigma}(\bar{X}_n - \mu)\right] = \frac{n}{\sigma^2}\mathrm{Var}[\bar{X}_n - \mu] = \frac{n}{\sigma^2}\mathrm{Var}[\bar{X}_n] = 1$$

Informally, the central limit theorem states that if a set of data is influenced by many small, random effects, then you will get a normal distribution.

This is why the Gaussian appears so much in nature!

We can apply the CLT

Application:

Let $X_1, \dots, X_n$ be iid RVs. Define $\bar{X}_n$ & Z_n as before.
Let Φ be the CDF for $N(0,1)$. Then

$$\lim_{n\to\infty} P[Z_n \le t] = \Phi(t)$$

Or more usefully

$$P[Z_n \le t] \approx \Phi(t) \text{ for large } n$$

Example:

Consider the Bernoulli distribution w/ $p(\text{Heads}) = 0.35$, that is RV $X \sim \text{Ber}(0.35)$. Estimate the results for 100 flips.

$E[X] = 0.35$

$\text{Var}[X] = \sqrt{p(1-p)} \approx 0.477$

$\frac{\sqrt{n}}{\sigma} \approx 21$

What is the probability of getting over 40 Flips?

$$P(\#H > 40) \approx P(\bar{X}_n > 0.4)$$
$$\approx P(\bar{X}_n - \mu > 0.05)$$
$$\approx P(Z_n > 21 \cdot 0.05)$$
$$\approx P(Z_n > 1.05)$$
$$\approx 1 - \Phi(1.05) \leftarrow \text{1 std above mean}$$

By the 68-95-99.7 rule, the approximate error is 16%.

Note: We made a bunch of approximations for this fairly easy problem. In practice you'd never do this for this particular problem but it's an easy first example.

Example: Continuity Correction

Flip n=100 coins. Let S = # heads be a RV.
Estimate $E[45 \le R \le 55]$.

Let X = Bernoulli(1/2). $\sigma[X] = \sqrt{V[X]} = \sqrt{p(1-p)} = 1/2$

$\sigma[X] = \sqrt{V[X]} = \sqrt{p(1-p)} = 1/2$

$\sigma[S] = \sqrt{np(1-p)} = \frac{\sqrt{100}}{2} = 5$

$E[S] = np = 50$

Let $X \sim N(\mu=50, \sigma=5)$. We use X to estimate S.

However, we have an issue
$P[45 \le S \le 55] = P[44 < S < 56]$ but $P[45 \le X \le 55] \neq P[44 < X < 56]$.

This arises b/c we're using a continuous RV to estimate a discrete RV. To correct for this we use
$P[44.5 < X < 55.5] \approx P[45 \le S \le 55]$.

We then use the CDF F_X of X to find our probability.
$P[45 \le S \le 55] \approx P[44.5 < X < 55.5] = F_X(55.5) - F_X(44.5)$.

Confidence Intervals

Consider $X_1, \dots, X_n$ iid RVs ($X \sim X_1 \sim \dots \sim X_n$). Let $\bar{X}_n = \frac{X_1 + \dots + X_n}{n}$. We want Our goal is to find $E[X]$.

Now take random samples $x_1, \dots, x_n$. Our sample mean $\hat{\mu}$ & sample variance $\hat{\sigma}^2$ is

$$\hat{\mu} = \frac{x_1 + \dots + x_n}{n}$$

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{\mu})^2$$

$$\hat{\sigma} = \sqrt{\hat{\sigma}^2} \quad \text{or} \quad \frac{1}{n-1}, \text{ look up "Bessel's correction"}$$

The approximate standard deviation of $\bar{X}_n \approx N(\mu, \sigma^2/n)$ is called the standard error $\bar{S}$ & is calculated

$$\bar{S} = \frac{\hat{\sigma}}{\sqrt{n}}$$

This standard error gives us a confidence interval for μ.
In particular, by the central limit theorem of 95% confidence interval states that μ falls w/in $[\hat{\mu} - 1.96\bar{S}, \hat{\mu} + 1.96\bar{S}]$ w/ 95% confidence.

We can generalize this by replacing 1.96 w/ the z-score of the $\frac{1-c}{2}$ percentile where c is our confidence percentage.

We state that in 95% of repeated experiments, the true mean μ would fall inside of the 95% confidence interval.

Monte-Carlo Integration

Consider a RV X w/ PDF f_X. Then $E[g(X)] = \int_{-\infty}^{\infty} g(t) f_X(t) dt$. ← This is a given we can assume
Thus to find $E[g(x)]$ we need to solve an integral.

However, in general integrals are hard to solve so how do we estimate $E[g(x)]$.

Consider the special case of $X \sim \text{Unif}(a, b)$.
$$\int_a^b g(x) dx = (b-a) \int_a^b g(x) \frac{1}{b-a} dx = (b-a) \int_a^b g(x) f_X(x) dx = (b-a) E[g(X)] \text{ where } X \sim \text{Unif}(a, b).$$

Monte Carlo is generally good b/c it's easy & efficient to apply to higher dimensions b/c it does not scale w/ dimension in difficulty.

However, quadrature (e.g. left/right riemann sums, trapezoidal rule, etc.) is generally much more accurate & efficient at small dimensions. However, it scales exponentially w/ the number of dimensions & so is intractable for higher dimensions.

Recall: Quadrature is finding the are under a curve by approximation e.g. Riemann sums, trapezoidal rule, simpson's rule, etc.

Note: By dimensions we mean free variables in the differential equation.

Remark: What is popular to call "the butterfly-effect", we call an ill-conditioned function.

Remarks:

Monte Carlo performs better for low variance functions.

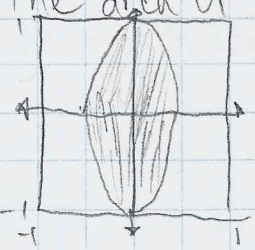
Monte Carlo accuracy scales w/ $\sqrt{n}$ (b/c standard error scales w/ $\frac{1}{\sqrt{n}}$).

To improve Monte Carlo accuracy, we can use control variates. Control variates is a method of reweighting your inputs to handle issues w/ sampling (e.g. undersampling women). Essentially, you're making your function lower variance to make Monte Carlo more accurate.

Goal: Given some function $g(x)$ which we want to approximate w/ $\tilde{g}(x)$, define a similar function $h(x)$ but simpler (so we can compute $E[h(x)]$).

Example:

Let's approximate the area of the ellipse $S = \{2x^2 + y^2 \leq 1\}$



Linear Algebra

Notation:

We will use householder notation where
- $A, B, C, \ldots$ are matrices
- $a, b, c, \ldots$ are vectors
- $\alpha, \beta, \gamma, \ldots$ are scalars

Def:

Two matrices $A, B \in \mathcal{M}^{n \times m}$ are equal iff $a_{ij} = b_{ij} \quad \forall i \in \{1, \ldots, n\} = [n] \;\&\; j \in [m]$

Def:

Given a matrix $A \in \mathcal{M}^{n \times m}$. The transpose of the matrix is a $A^T \in \mathcal{M}^{m \times n}$ where

$A^T = [a_{ji} \quad \forall i \in [n] \;\&\; j \in [m]]$.

A matrix is symmetric iff $A^T = A^T$.

Def:

Let $A \in \mathcal{M}^{n \times m}$ & $B \in \mathcal{M}^{m \times l}$. We define the product $C \in \mathcal{M}^{n \times l}$ as

$C = AB = [c_{ij} = \text{row}(A, i) \cdot \text{col}(B, j) \quad \forall i \in [n] \;\&\; j \in [m]]$

Example:

$$B = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix}$$

$$A \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \\ a_{41} & a_{42} \end{bmatrix} \to \begin{bmatrix} 0 & \downarrow & \\ & & \\ & \to 0 & \end{bmatrix}$$

Note: Matlab uses column-major order!

Rem:

Left multiplication by a diagonal matrix scales the rows.
Right multiplication by a diagonal matrix scales the columns.

Def:

$A \in \mathcal{M}^{n \times n}$ is invertible or non-singular iff $\exists B \in \mathcal{M}^{n \times n}$ s.t. $AB = BA = I$

We say $A^{-1} = B$.

A is singular iff it is not invertible.

Def:
A matrix $Q \in \mathbb{M}^{m \times n}$ is orthogonal iff
$QQ^T = Q^TQ = I$ or equivalently $Q^T = Q^{-1}$

This is true if each row/col of Q has a 2-norm equal to one
&
$$q_i^T q_{ij} = \begin{cases} 1 & i=j \\ 0 & o/w \end{cases}$$
Where $Q = [a_1, \dots, a_n]$

Example:
$Q = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ is orthogonal.

Def:
A matrix $Q \in \mathbb{M}^{m \times n}$ where $n < m$ is called orthonormal iff
$Q^TQ = I_n$

Example:
$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$ is orthonormal

$Q^TQ = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $QQ^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Thm:
If $Q \in \mathbb{M}^{m \times n}$ is orthonormal, then

$$\|Qx\|_2 = \|x\|_2 \quad \forall x \in \mathbb{F}^n$$

Intuitively, orthonormal matrices can reflect & rotate, but not distort.

Matrix Norms

How can we tell how close a matrix is to being invertible? The answer is norms.

The reason we want this tolerance is b/c in real life we get rounding error.

Def:
A matrix norm $\|\cdot\|$ is a function that satisfies the following properties
i) Positivity: $\|A\| \geq 0 \quad \forall A \in \mathbb{R}^{m \times n}$
ii) Positive definite: $\|A\| = 0$ iff $A = 0$
iii) Absolute scaling: $\|\alpha A\| = |\alpha| \|A\| \quad \forall \alpha \in \mathbb{R}$ & $A \in \mathbb{R}^{m \times n}$
iv) Triangle Inequality: $\|A+B\| \leq \|A\| + \|B\| \quad \forall A, B \in \mathbb{R}^{m \times n}$

Some people also demand submultiplicativity
* v) Submultiplicativity: $\|AB\| \leq \|A\| \|B\|$

Def: An operator norm or matrix 2-norm depends is defined as

$$\|A\|_2 = \max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2}$$

Intuitively, it's the largest factor by which a linear transformation defined by A may stretch a vector.

Def: The Frobenius norm is given by

$$\|A\|_F = \sqrt{\sum_{i,j} |a_{ij}|^2}$$

This essentially treats the matrix as one long mn vector & takes the 2-norm.

Thm: The Operator norm & Frobenius norms are submultiplicative. That is

$$\|AB\|_2 \leq \|A\|_2 \|B\|_2 \quad \& \quad \|AB\|_F \leq \|A\|_F \|B\|_F$$

Thm: Let $r = \text{rank}(A)$. Then

$$\|A\|_2 \leq \|A\|_F \leq \sqrt{r}\|A\|_2$$

Thm: If P & Q are orthogonal matrices then

$$\|PAQ^T\|_2 = \|A\|_2 \quad \& \quad \|PAQ^T\|_F = \|A\|_F$$

That is these norms do not change under orthogonal transformation.

To reframe the question of being close to being singular, Given a matrix A where $\text{rank}(A) = r$. What is the $k < r$ rank matrix closest to A (using norms)?

We find this matrix by Singular Value Decomposition (SVD).

Singular Value Decomposition

Def: Let A be a $m \times n$ matrix. A singular value decomposition (SVD) of A is matrices $U \in \mathbb{R}^{m \times r}$ $\Sigma \in \mathbb{R}^{r \times r}$ & $V \in \mathbb{R}^{n \times r}$ where U & V are orthogonal & $r = \text{rank}(A)$ such that (next page)

such that

$A = U \Sigma V^T$ where U & V are orthogonal

& $\Sigma = \begin{bmatrix} \sigma_1 & & 0 \\ & \sigma_2 & \\ & & \ddots \\ 0 & & \sigma_r \end{bmatrix}$ w/ $\sigma_1 \geq \sigma_2 \geq \cdots > \sigma_r \geq 0$ ← diagonal matrix

The columns of U are the <u>left singular vectors</u> & columns of V the <u>right singular vectors</u>.

The diagonal entries of Σ $\sigma_1, \ldots, \sigma_n$ are called the <u>singular values</u>.

Note: Alternatively, instead of r we could use $s = \min(n, m)$. In this case $\sigma_1 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_s = 0$ & vectors $v_{r+1}, \ldots, v_s$ are null vectors of A.

<u>Rem:</u>

If U & V have columns u_i & v_i, then we could write the matrix A is <u>outer product form</u> that is

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

Since $\|A\|_F^2 = \sigma_1^2 + \cdots + \sigma_r^2$ we can think of A as a sum of rank 1 matrices where the ones w/ the greatest singular values contribute the most mass to A.

<u>Thm:</u>

The SVD satisfies the following properties

i) The rank of X is the number of non-zero singular values

ii) $\|A\|_2 = \sigma_1$

iii) If A is invertible, then $\|A^{-1}\|_2 = 1/\sigma_1$

iv) σ_r is the 2-norm (& F-norm) of the smallest perturbation E such that $A+E$ is singular

The optimal perturbation is $E = -\sigma_r u_r v_r^T$

v) $\|A\|_F = \sqrt{\sum_{i=1}^{r} \sigma_i^2}$

vi) If A_k is best rank-k approximation of A, then

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$$

vii) The projection onto the best k-dimensional subspace is given by:

$$\Pi_k = U_k U_k^T \quad \text{where} \quad U_k = [u_1 \ldots u_k]$$

How do we find the condition number of solving $Ax = b$?

Our input is $\tilde{b} = b + \delta b$ (where δ is our uncertainty).
Our output is $\tilde{x}$. We find the relative error by

CATCH UP ON NOTES

(Non) Uniqueness of SVD

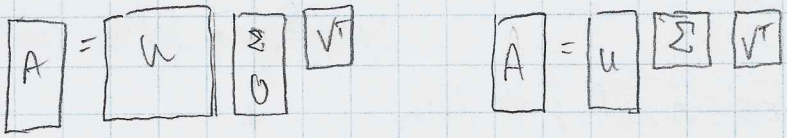
Thm:
The SVD of a matrix A always exists & is almost unique.

Example:

- Let $A = I$. Then $I = QIQ^T$ for any orthogonal Q.
- Let $A = U\Sigma V^T$. Then $A = (-U)\Sigma(-V)^T$ also.
- Let $A = U\Sigma V^T$ where Σ has repeated values (i.e. A has duplicate singular values). Then intuitively (is it really?) this gives us multiple options.
  what?!?

Thm:
Given any matrix, the singular values $\sigma_1, \ldots, \sigma_r$ are unique & so are span(left singular vectors) & span(right singular vectors).

SVD Recap



Full SVD "Economy" SVD

Truncated SVD

Def:
Consider $A \in \mathcal{M}^{m\times n}$. If $\text{rank}(A) = r > k$, then the k-rank truncated SVD of A is

$$\tilde{A}_k = \underset{m\times m}{U}\ \underset{m\times n}{\begin{bmatrix}\sigma_1 & & & \\ & \ddots & & \\ & & \sigma_k & \\ & & & 0\end{bmatrix}}\ \underset{n\times n}{V^T} = \underset{m\times k}{U}\ \underset{k\times k}{\begin{bmatrix}\sigma_1 & & \\ & \ddots & \\ & & \sigma_k\end{bmatrix}}\ \underset{k\times n}{V^T}$$

Example:
Let $A = \begin{bmatrix}0 & 0 & 2\\ -2 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix} \in \mathcal{M}^{4\times 3}$. The singular values are $\sigma_1 = 2$, $\sigma_2 = 2$, $\sigma_3 = 1$.

Find the 2-rank truncated SVD (i.e. $\tilde{A}_2$).

$$\tilde{A}_2 = \begin{bmatrix}-1 & 0\\ 0 & 1\\ 0 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}2 & 0\\ 0 & 2\end{bmatrix}\begin{bmatrix}0 & -1\\ 0 & 0\\ -1 & 0\end{bmatrix}^T = \begin{bmatrix}0 & 0 & 2\\ -2 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}$$

The k-rank truncated SVD can be viewed as an approximation of A, or the closest k-rank matrix to A. (see later theorem)

Thm:
$\tilde{A}_k$ is unique iff $\sigma_k < \sigma_{k+1}$.

Thm:
If $\text{rank}(A) = r > k$ then $\tilde{A}_k$ minimize $\|X - A\|_2$ & $\|X - A\|_2$ over $\text{rank}(X) \le k$.

Note that
$\|\tilde{A}_k - A\|_2 = \sigma_{k+1}$ & $\|\tilde{A}_k - A\|_F = \sqrt{\|A\|_F^2 - \|\tilde{A}_k\|_F^2}$

Example:
From our earlier example

$$A = \begin{bmatrix}0 & 0 & 2\\ -2 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix} \quad \& \quad \tilde{A}_2 = \begin{bmatrix}0 & 0 & 2\\ -2 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix} \quad \& \quad \tilde{A}_1 = \begin{bmatrix}0 & 0 & 2\\ 0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix} \left(\text{or } \begin{bmatrix}0 & 0 & 0\\ -2 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}\right)$$

$\|\tilde{A}_2 - A\|_2 = 1$ & $\|\tilde{A}_2 - A\|_F = 1$ & $\|\tilde{A}_2 - A\|_2 = 2$ & $\|\tilde{A}_2 - A\|_F = \sqrt{2^2 + 1^2} = \sqrt{5}$

SVDs & Image Compression

Note: Optimal SVDs can be slow

Images can be viewed as a matrix of pixel data.

Storing this naively takes mn storage. However, if we store it in factored form as a k-truncated SVD, then it takes $(m+n)k$ storage.

We can determine the error b/w the real matrix using Matrix norms.

relative error $= \frac{\|\tilde{A} - A\|_F}{\|A\|_F}$ absolute error $= \|\tilde{A} - A\|_F$ (could use 2-norm)

Programming Tools & SVDs

How do we find the SVD in MatLab?

1) [U, S, V] = svd(A, 'econ') then truncate
   Pro: optimal
   Con: slow if A is large
2) [U, S, V] = svds(A, k) (rank k approximation)
   Pro: good if $k << \min(m, n)$ esp is A is sparse
   Con: slow if k is large
3) svdsketch(A, τ) (τ for tolerance)
   $\hat{A} \approx \hat{A}_k$ b/c $\|\hat{A} - A\|_F < \tau \|A\|_F$
   Pro: fast, esp. w/ large A.
   Con: not optimal

SVDs & Dimension Reduction

Dimension reduction is still very abstract, but in general the problem is, given a large amount of data, how do you optimally describe it w/ few data points.

For example, given the voting history of politicians, how do we "score" them along some small collection of "axises."

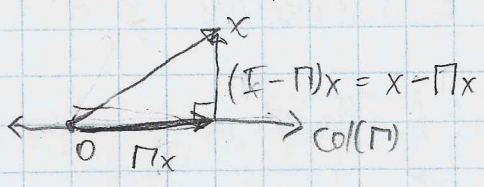
It can also be used to classify groups, for example as you do w/ the Fisher Iris Dataset.

Orthogonal Projections

Def:
An <u>orthogonal projection matrix</u> Π is a matrix where

$$\Pi = \Pi^T \quad \& \quad \Pi^2 = \Pi$$

Note that Π is NOT an orthonal matrix but instead Πx projects x orthogonally onto range(Π) = col(Π).



Thm:
If Q is a basis (matrix) for $\text{col}(\Pi)$, then $QQ^T = \Pi$.

Thm:
If Π is an orthogonal projection matrix, then its singular values are either 0 or 1.

Principle Component Analysis (PCA)

We can use SVDs to approximate datapoints. However, SVD requires that the approximation pass thru 0, even when

Side Note

Def:
A matrix is said to be badly conditioned (aka not-invertible) if it is nearly singular, in other words the ratio b/w it's large singular values & small ones is extremely large.

In other words, if σ_1/σ_n is extremely large, then the matrix is badly conditioned.

Rem:
All orthogonal matrixes are well conditioned.

Rem:
An orthogonal matrix only rotates the entire space, no scaling.

Note: The SVD (& principal component analysis) are great for information compression but aren't very explainable/interpretable.

Denoising (Images)

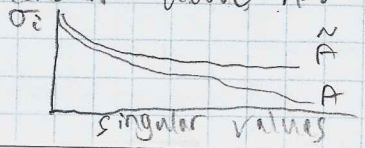
Say we're collecting some data (e.g. an image) but have some signal noise or missing data. How do we recover the missing/covered information?

Here's some common types of noise
- Gaussian Noise: Every single pixel gets shifted by an amount described by a normal distribution centered at 0. (i.e. gets a little darker/brighter)
- Salt & Pepper Noise: Some pixels get set to 0 or 1 randomly

The main idea is to use our truncated SVD again.

Let $\tilde{A} = A + E$ where A is the original image, E is noise/error, & $\tilde{A}$ is the corrupted image.

Since we assume that noise is evenly distributed, $\tilde{A}$'s singular values are some constant above A's.



For the largest singular values of $\tilde{A}$, the amount of error is small relative to the actual information (i.e. a high signal to noise ratio).

Thus our SVD should give us a good image.

However, in practice it struggles by itself b/c the Frobenius norm doesn't agree well w/ human intuition on how close images are.

We can improve this w/ blurring, also called inverse problems.

Inverse Problems

We want to find the inverse of some vector b but we only have a noisy vector $\tilde{b} = b + e$

$$Ax = \tilde{b}$$
$$\Rightarrow x = A^{-1}\tilde{b} = A^{-1}(b+e)$$

We can solve this by first blurring A, which can be modeled by a linear operator / matrix. Essentially, you define a matrix B_r & B_c which blur the rows & columns respectively. Then

$$\text{blurred } A = B_c A B_r$$

Since we can define blurring as linear operators, we can unblur w/ their inverses.

The issue is these blurring matrices are ill-conditioned. Intuitively, it's hard to tell the difference b/w a blurred black & white checkerboard & a gray sheet.

So $x = A^{-1}\tilde{b} = A^{-1}b + \underbrace{A^{-1}e}_{\text{may overpower } A^{-1}b}$

Rem:

Most blurring is done w/ Gaussian blurring. Basically you describe your blurring where you sum up your nearby pixels w/ weights described by the PDF of the normal distribution.

Theoretically, you'd have to add all pixels together, but pixels beyond a few standard deviations are negligible.

When we naively unblur the image, we get results ruined by noise. Basically we get abstract patterns.

We can get around this horrible pattern mess by using pseudo-inverses of a k-rank truncated SVD.

Def:
Let A be a matrix w/ SVD $A = U\Sigma V^T$.

The pseudo-inverse of A, that is A^* is

$$A^* = V\Sigma^{-1}U^T$$

Basically, it's the closest we can get to the inverse when the matrix isn't formally invertible.

If we take the k-rank truncated SVD of A & take it's pseudo-inverse, we then just need to tune k so that we get as much signal in the inverse but only a small amount of noise.

Random Variables

I missed last class so I am missing notes on independence.

Recall: Independence

Two random variables X & Y are independent iff

$$F_{XY}(x,y) = F_X(x) F_Y(y)$$

or equivalently

$$Pr(X \le x \ \& \ Y < y) = Pr(X \le x) Pr(Y \le y).$$

Also $E[XY] = E[X]E[Y]$

Intuitively, X & Y give no info about each other.

Similarly, if the PDFs exist then

$$f_{XY}(x,y) = f_X(x) f_Y(y).$$

The method of least-squares w/ outputs y, inputs X, & coefficients $\beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \end{bmatrix}$ tries to find

$$\min_{\beta} \|y - X\beta\|_2$$

each row is a datapoint

In other words we want to find the best coefficients such that $y \approx X\beta$.

Recall: Control Variates

We have some control variate X w/ a known $E[X]$ & we want to use it to find $E[Y]$ for some Y.

Strategy: Let $Z = Y - \alpha(X - E[X])$.

We then find $\alpha \in \mathbb{R}$ such that

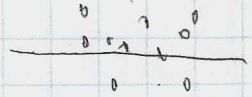
$$\min Var[Z] = \min Var[Y - \alpha X].$$

Def:

Covariance measures the linear relationship b/w RVs X & Y.

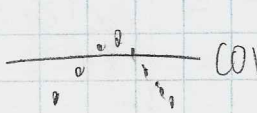
$$Cov[X,Y] = E[(X - \mu_X)(Y - \mu_Y)] = E[XY] - E[X]E[Y].$$

What do we mean linear relationship? Basically the slope of the best fit line using samples of (X, Y)



$Cov[X,Y] = 0$

$X \sim Y \sim Unif(0,1)$ ← independent



$Cov[X,Y] = 0$ ← not independent

$X \sim Unif(0,1)$ $Y = (X - 0.5)^2$

Thm:
Let X & Y be independent RVs. Then $COV[X,Y]=0$.

Pf:

$$\begin{aligned} COV[X,Y] &= E[(X-\mu_X)(Y-\mu_Y)] \\ &= E[X-\mu_X]\,E[Y-\mu_Y] \\ &= (E[X]-\mu_X)(E[Y]-\mu_Y) \\ &= 0. \end{aligned}$$

Note that the converse doesn't hold. (Shown earlier)

Thm:
Here's some properties of covariance. Let X, Y, Z be RVs & $\alpha, \beta \in \mathbb{R}$.

- $COV[\alpha, X] = 0$
- $COV[X, X] = Var[X]$
- Bilinearity: $Cov(\alpha X + \beta Y, Z) = \alpha COV(X, Z) + \beta COV(Y, Z)$
- Symmetric: $COV[X, Y] = COV[Y, X]$.
- (Almost) Positive Definite: $COV[X, X] = Var[X] \geq 0$ & $COV[X, X] = 0$ iff X = constant

This means covariance is (almost) an inner product so we can do linear algebra, like the 2-norm!

Example:
Let $U_1, U_2 \overset{iid}{\sim} Unif(1, -1)$.
Let $X = 2U_1 + U_2$ & $Y = U_1 - 2U_2$.

Find $Cov(X, Y)$.

$$\begin{aligned} Cov(X,Y) &= Cov(2U_1 + U_2,\ U_1 - 2U_2) \\ &= Cov(2U_1, U_1) \\ &\quad + Cov(2U_1, -2U_1) \quad \text{independent} \\ &\quad + Cov(U_2, U_1) \\ &\quad + Cov(U_2, -2U_2) \\ &= 2Var(U_1) - 2Var(U_2) \quad \text{both have some variance } \sigma^2 = Var(U_1) = Var(U_2). \\ &= 2\sigma^2 - 2\sigma^2 \\ &= 0. \end{aligned}$$

Going back to control variates, how do we find the optimal α?

That is, given $Z = Y - \alpha X$, find α that minimizes

$$\begin{aligned} Var[Z] &= Cov(Y - \alpha X,\ Y - \alpha X) \\ &= Var(Y) - 2\alpha Cov(X, Y) + \alpha^2 Var(X) \end{aligned}$$

This gives us

$$\alpha = \frac{Cov(X, Y)}{Var(X)}.$$

Plugging in α gives us

$$Var[Z] = Var[Y] - \frac{Cov(X, Y)^2}{Var(X)} \Rightarrow \frac{Var(Z)}{Var(Y)} = 1 - \boxed{\frac{Cov(X, Y)^2}{Var(X)Var(Y)}}$$

ρ^2

Optimization

How do we maximize/minimize some objective function $f(x)$.

Example:

Given a data matrix $X \in \mathbb{R}^{n \times d}$ (n points & d dimensions), what k-dimensional subspace best approximates the data?

Scoring matrix approximations using 2-norm or Frobenius norm, formally

$$\max_{\text{rank}\,\Pi = k} \|X\Pi\|_{2 \text{ or } F}$$

$$\min_{\text{rank}\,\Pi = k} \|X(I - \Pi)\|_{2 \text{ or } F}$$

The Eckart-Young theorem says this is the rank-k truncated SVD.

Problem: Line Fitting

Given some set of datapoints, what (straight) line best approximates the data?

How do we score a line? Formally, we describe our lines as

$y \approx \alpha + \beta x$ ← real data

$\hat{y} = \alpha + \beta x$ ← prediction

Idea 1:

Minimize the sum of distances b/w actual & predicted values. That is minimize

$$\sum_i |\hat{y}_i - y_i| = \sum_i r_i \quad \text{where } r_i = |\hat{y}_i - y_i| \text{ is the residuals}$$

In terms of matrices. Let

$$b = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \qquad A = \begin{bmatrix} 1 & t_1 \\ \cdots & \\ 1 & t_n \end{bmatrix} \qquad x = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

b: actual data; Ax: predicted

We want $b \approx Ax$. Let $r = Ax - b$ (so $r_i = |\hat{y}_i - y_i|$): residual vector

We want

$$\min_{x \in \mathbb{R}^2} \|Ax - b\|_1 = \min \sum_i r_i.$$

Idea 2:

Let's minimize the squared residuals.

Why? It makes the math work out nicer & "punishes" being really far off.

W/ $b, A, x,$ & r defined as earlier, we want $\min_{x \in \mathbb{R}^2} \|Ax-b\|_2^2 = \min \sum r_i^2$.

Def:

Let $A \in \mathbb{R}^{m \times n}$ w/ $b \in \mathbb{R}^m$ & $x \in \mathbb{R}^n$. Our optimization problem, $b = Ax$ is typically impossible b/c the system is normally over determined b/c $m > n$.

Approach 2 ← I zoned out for approach 1 :/

This approach uses calculus. Let $f(x)$ be our objective function

$f(x) = \|Ax-b\|_2^2 = (Ax-b)^T(Ax-b) = x^TA^TAx - 2b^TAx + \underbrace{b^Tb}_{\text{constant}}$

Say x is a solution to $\min f(x)$. Then

$A^T(b-Ax_0) = 0$

$\Rightarrow (Az)^T(b-Ax_0) = 0 \quad \forall z \in \text{col}(A)$

Applying the Pythagorean theorem, we get for any $x \in \mathbb{R}^n$

$\|Ax-b\|^2 = \|A(x-x_0) + (Ax_0 - b)\|_2^2$

$= \underbrace{\|A(x-x_0)\|_2^2}_{\text{"penalty" for distance of } x \text{ from } x_0} + \underbrace{\|b-Ax_0\|_2^2}_{\text{optimal residual}}$

Say x_* solves $\min \|Ax-b\|$. Then x' is also a solution

$A(x'-x_*) = 0$

Thus the set of solutions is $x_* + \text{null}(A)$

General Linear Algebra Review

Def:

A projection Π onto space $U \subseteq V$ is a linear transformation such that

$$\Pi v = \begin{cases} v & \text{iff } v \in U \\ 0 & \text{iff } v \in U^\perp \end{cases}$$

Thm:

For all projections Π, $\Pi^2 v = \Pi v$ (so $\Pi^n = \Pi \ \forall n \in \mathbb{N}$)

Thm:

Let Q be an orthogonal basis for subspace $U \subseteq V$. Then

$\Pi_U = QQ^T$

Residuals & r^2

Given RVs X & Y their r^2 coefficient is

$$r^2 = \frac{Cov(X,Y)^2}{Var(X)\,Var(Y)} \leq 1$$

Note that

$$Cov(X,Y)^2 \leq Var(X)\,Var(Y) \Rightarrow \underbrace{Cov(X,Y) \leq \sigma_X \sigma_Y}$$

Cauchy-Swarz Inequality
$\|x^T y\|_2 \leq \|x\|_2 \|y\|_2$

Def:
Given RVs X & Y define the correlation of (X,Y) is

$$r = \frac{Cov(X,Y)}{\sigma_X \sigma_Y}.$$

or $\rho(X,Y)$

In the law of cosines
$v \cdot u = \|v\|_2 \|u\|_2 \cos(\theta)$
r is like $\cos(\theta)$.

For example $-1 \leq r \leq 1$ & $|r| = 1 \Rightarrow$ perfect linear relation b/w X & Y.

Thus in terms of stats $0 \leq r^2 \leq 1$ describes what fraction of the variation in Y is explained by X.

$$\frac{Var(Y - \alpha X)}{Var(Y)} = 1 - r^2$$

Examples:
Let $X \sim Unif(0,5)$ & $Y = X/10 + N(0,1)$

If we calculate the correlation coefficients we get
$r \approx 0.1$ & $r^2 \approx 0.01$.
We say X explains 1% of the variation in Y.

The relation to linear fits is

$$\alpha = \frac{Cov(X,Y)}{Var(Y)} \quad \& \quad r = \frac{Cov(X,Y)}{\sigma_X \sigma_Y}.$$

Note that r is unitless & invariate to scaling. That is
$r(\alpha X + \beta, \gamma Y + \delta) = r(X,Y)$ when $\alpha \neq 0$ & $\gamma \neq 0$.

This means °F & °C doesn't change anything.

α, the slope of the best fit line is equal to r if X & Y are normalized so $\sigma_x = \sigma_y$.

Remark:
If $x_1, \dots, x_n$ from X & $y_1, \dots, y_n$ from Y where X & Y are RVs, then

$$\mathrm{Cov}(X,Y) \approx \underbrace{\frac{1}{n}}_{\text{or } \frac{1}{n-1}} \sum_{i=1}^{n} (x_i - \mu_x)(y_i - \mu_y)$$

Covariance Matrix

We want to compare MATLAB's bit strings & our bit strings. (Or coin flips)

Our strategy is to view the correlation b/w flip N & flip $N+1$. We can create confusion matrixes

Note: All eigenvalues of a covariance matrix are non-negative

(N+1)/Y	T	H
T	304	315
H	315	318

(N/X)

X = Flip N
Y = Flip $N+1$
$T = 0$ & $H = 1$
} both are Bernoulli

We try to find the covariance b/w X & Y.

$$P(X=H) = P(Y=Y) = \frac{633}{1252} = E[X] = E[Y]$$

$$\mathrm{Cov}(X,Y) = E[XY] - E[X]E[Y] = \frac{318}{1252} - \left(\frac{633}{1252}\right)^2 = -0.0016$$

Since the covariance is low, we expect this to be MATLAB's random code.

If we got worse, then we'd get that the bitstrings often alternate more than they should. We'd say this is likely the student-generated one.

Random Vectors

Ref:
A random vector $\vec{X} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$ is a vector whose entries are RVs.

Note: $\mu_{\vec{x}} = E[\vec{X}] = \begin{bmatrix} E[X_1] \\ \vdots \\ E[X_n] \end{bmatrix}$.

How do we define variance? variance defined componentwise is limiting.

Def: The Covariance matrix K_{xx}

$$K_{xx} = \begin{bmatrix} \mathrm{Var}(X_1) & \cdots & \mathrm{Cov}(X_1, X_n) \\ \vdots & \ddots & \vdots \\ \mathrm{Cov}(X_n, X_1) & \cdots & \mathrm{Var}(X_n) \end{bmatrix}$$

↑ or $\mathrm{Cov}(X_n, X_n)$

Properties:

- $(K_{xx})_{ij} = \mathrm{COV}(X_i, X_j)$
- K_{xx} symmetric $\Leftrightarrow K_{xx} = K_{xx}^T$ b/c $\mathrm{Cov}(\cdot,\cdot)$ symmetric
- $K_{xx} = E[XX^T] - \mu_x \mu_x^T$

Thm: If $Y = a^T X = \sum_{i=1}^{n} a_i X_i$ then $\mathrm{Var}[Y] = a^T K_{xx} a$.

Coro:

$a^T K_{xx} a = \mathrm{Var}[a^T X] \geq 0$

So K_{xx} is positive semidefinite (semidefinite b/c there may be $a \neq 0$ s.t. $a^T X = 0$)

Example:

Let $\mathrm{Var}(X_1) = 1$, $\mathrm{Var}(X_2) = 2$, $\mathrm{Cov}(X_1, X_2) = 1$ & $Y = 3X_1 - X_2$

$$\begin{aligned} \mathrm{Var}(Y) &= \mathrm{Cov}(3X_1 - X_2,\ 3X_1 - X_2) \\ &= 9\,\mathrm{Var}(X_1) - 6\,\mathrm{Cov}(X_1, X_2) + \mathrm{Var}(X_2) \\ &= 9 - 6 + 2 \\ &= 5 \end{aligned}$$

} Properties of $\mathrm{Cov}(\cdot)$

Alternatively,

$$K_{xx} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$$

so

$$\mathrm{Var}(Y) = \begin{bmatrix} 3 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 3 \\ -1 \end{bmatrix}$$

$$= \begin{bmatrix} 3 & -1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$= 5$$

Multivariate PDFs

Def

A multivariate PDF $f_X(x_1, \dots, x_n) : \mathbb{R}^n \to \mathbb{R}$ has the properties

$f \geq 0$

$\int\int\int_{\mathbb{R}^n} f = 1$

In the 1-dimensional case we had simple intervals

$Pr[a \leq X \leq b] = \int_a^b f(t)\,dt$

In the n-dimensional case, we define subsets $\Omega \subseteq \mathbb{R}^n$. Then

$PR[X \in \Omega] = \int\int\int_{\Omega} f(t)\,dt.$

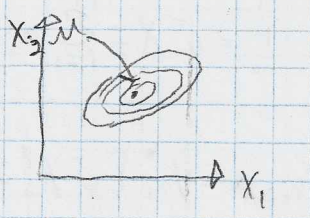
Def:

Generalization of Gaussian distribution

X is a normal random vector

$X \sim N(\mu, \Sigma)$ where $\mu \in \mathbb{R}^n$ & $K \in \mathbb{R}^{n \times n}$

$$f_X(x_1, \dots, x_n) = \frac{1}{\sqrt{(2\pi)^n \det(K)}} e^{-(x-\mu)^T K^{-1}(x-\mu)/2}$$



We say $X_1, \dots, X_n$ have a joint Gaussian distributions.

Here we assume K^{-1} exists. Not everyone does this.

These look like ellipsoids. No matter what "angle" you look at the distribution you get a Gauss

This things relation to principle component analysis (PCA) / singular vectors $v_1, \dots, v_n$ is that v_1 gives the l.c. w/ the greatest variance & v_n the least.

Remark:

$X \sim N(0, I_n)$ is n iid RVs $X_1, \dots, X_n \sim N(0,1)$.

Thm:

Let $X \sim N(0, I_n)$ X is rotationally invariant, that is Let Q be orthogonal & Let $Y = QX$. Then $Y \sim N(0, I_n)$.

Here's the Proof:

$$\begin{aligned} K_{YY} &= Q K_{XX} Q^T \\ &= Q I_n Q^T \\ &= QQ^T \\ &= I_n. \end{aligned}$$

Thm:

Let X be a random vector & A be a linear transformation.

Let $K_{XX} = Cov(X)$. Let $Y = AX$. Then

$Cov(Y) = K_{YY} = A K_{XX} A^T$.

k-Means Clustering

Given data $S = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$, we want to split the data into k-clusters defined by means $M = \{\mu_1, \dots, \mu_k\}$. (A)
The defines a partition (B)
$S = S_1 \cup \dots \cup S_k$.

The "best" clustering minimize some function f.
$\min_{\mu, P} f(\mu, P)$.

The traditional function is the sum of squared means.
So

$$\min_{|\mu|=k} \sum_{i=1}^{n} \min_{1 \le j \le k} \|x_i - \mu_j\|_2^2$$

$\{\mu_1, \dots, \mu_k\}$: every mean
$\min_{1 \le j \le k}$: The closest mean

This is cool, but how do we find the right means?

Algorithm:
The way we do this is by alternating b/w (A) partitioning the data & (B) computing the means iteratively.

0) Let k be the number of means.
1) Choose k-means arbitrarily (such that at least 1 point in each part)
2) Partition data based on proximity to means
3) Update the means of each partition based on their actual mean
4) Go to step 2 & repeat until means are stabilized

Pros: Cheap, sum-of-squares distance decreases w/ each step, will eventually converge to a solution

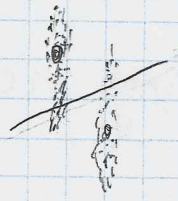
Cons: Might not find the optimal solution, (gets stuck at local mins) not always what you want

Note that k-means clustering breaks the space into Voronoi cells w/ centers of the means.

When does k-means clustering fail?
1) Differently sized (or spread) clusters get treated as "same size"

← "Mickey Mouse Issue"

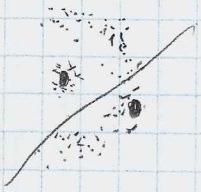
2) Oblong / not close to circular. You'll accidentally misclassify the tips



Luckily we can fix this by making every axis the same variance

3) You have vastly different populations, that is one dataset has 100,000 points & one has 100. You'll split the larger ones.

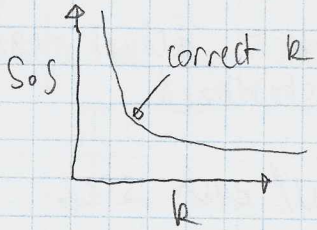
4) Non-Gaussian data. You just can't split non-blobs w/ Voronoi cells



There's another more logistical issue. How do you choose k if you don't already know it?

We can't just look at the sum of squares SoS distances b/c adding points always reduces it. (Consider a cell for every point.)

One way is to try multiple k & look at SoS distance. The correct answer should be a "kink" in the graph. That is, fewer means really hurts & more doesn't help much.



You could "just look" but that takes time & it's hard to look at 50-dimensional data.

Note that high-dimensional data suffers the "curse of dimensionality." That is distance behaves unintuitively. Basically, high dimensional data gets very far & nearly equally far apart. Distance gets "spikey!"

This means k-means starts to be less useful b/c it gets more sensitive to noise.

To fix the curse of dimensionality, we do principle component analysis (PCA) or singular value decomposition (SVD) to reduce the dimensionality before running k-means.

$\tau > 0 \Rightarrow$ more negatives $\tau < 0 \Rightarrow$ more positives

Def:
The tolerance τ using a linear model is used to set how conservative/liberal we are based by linear models. That is, w/ +1 being success & -1 as failure $\hat{y} = \text{sign}(\tilde{y} - \tau)$ where $\tilde{y} = \alpha_0 + \alpha_1 x_1 + \cdots + \alpha_d x_d$

Accuracy might not always be the best measurement, especially for rare events b/c you can just say the rare event never happens. For example, COVID tests.

Def:
Binary classification is trying to predict true/false conditions using some data.

For example, disease tests.

Def:
A confusion matrix is a way to measure how good a binary classification method is. It is a 2x2 matrix:

TN	FP
FN	TP

TN = True Negative FP = False positive
FN = False Negative TP = True Positive

We don't always consider all errors equally. Maybe a false positive isn't that bad but a false negative is really bad.

Def:
We have several measures based off the confusion matrix.

$$\text{accuracy} = \frac{TN + TP}{\text{total}}$$

$$\text{error rate} = \frac{FN + FP}{\text{total}}$$

$$\text{specificity} = \frac{TN}{TN + FN}$$

$$\text{precision} = \frac{TP}{TP + FP}$$

} There are others ⇓

called the decision boundary

Simple binary classifiers work by drawing a hyperplane in the space of all datapoints. This hyperplane begets a 1D line split in the middle & you can visualize the data by a histogram along the normal line.

If we add more parameters, we become more sensitive to noise as we begin to overfit the training data. This means we generalize better.

In general, to get around this we do PCA or SVD to reduce the dimensionality which can reduce overfitting.

Here is an expanded confusion matrix

actual \ predicted	+	−	
+	TP	FN	Recall/sensitivity TP/P
−	FP	TN	Specificity TN/N
	Precision $\frac{TP}{TP+FP}$	$\frac{TN}{TN+FN}$ ← Negative Predictive Value	Accuracy $\frac{TP+TN}{total}$

Remark: Bayes Rule
Let A & B be 2 random events,

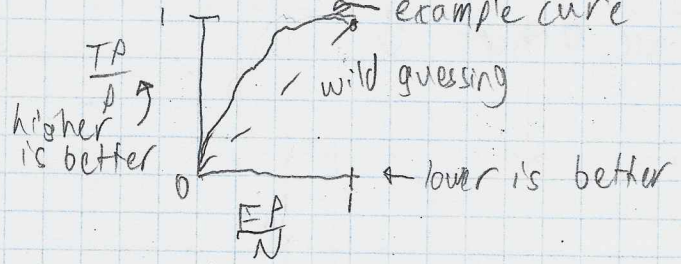
$$P(A \mid B) = \frac{P(B \mid A)\,P(A)}{P(B)}$$

You can also make a tree

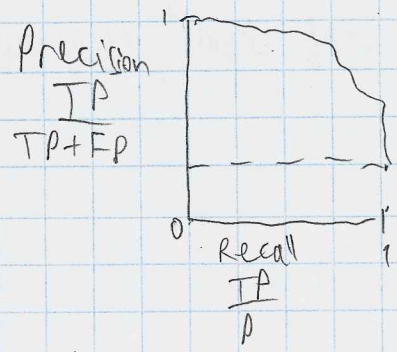
An implication is that an uncommon outcome for the majority often is more likely than a common outcome for a minority.

ROC & PR Curves

Def: Receiver Operating Characteristic (ROC)
The ROC curve shows the true positive rate vs the false positive rate for various values for x.



Def: Precision-Recall (PR)
The PR curve shows the precision vs recall for various values for τ.



Multi-class Classifiers

What if we try to classify things into k categories?
$y \in \{1, \ldots, k\}$.

One strategy is 1 vAll classifications. That is instead of having a more complicated classifier you have 3 binary classifiers that either say "yes" or "no" for every category. (It's actually more like a rating in each classifier, a rating b/w -1 & 1)

What do you do when there's no clear winner? That is 2+ are positive or all are negative? Generally you pick the largest in all cases.

KNN Classification

The idea w/ KNN is when you get a data point you haven't seen, look at the nearest k data points you "know" (or were trained on). Go w/ whatever the majority votes.

Normally you pick an odd k to avoid ties.

The issue here is how do you pick k? The larger k the smoother the decision boundary is.

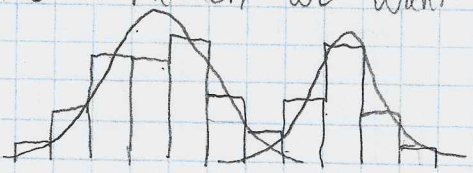
Pros: Works great w/ non-linear data, simple, easy to add new training points (unlike least squares), good w/ outliers, great w/ non-linear decision boundaries

Cons: Scaling can cause misclassifications, large & slow (esp in higher dims), weak to noise, need whole dataset, suffers from curse of dimensionality (use PCA!)

Gaussian Mixture Models

Give some data set $S = \{x_1, \dots, x_n\} \subset \mathbb{R}$. How can we describe S as a collection of Gaussian distributions?

That is, where $N_1, \dots, N_k$ are random variables, $N_1 + \dots + N_k$ should accurately describe S. Further, we want k to be small.



How do we formally fit the model?

Def:

Let $X = \rho_1 N_1 + \dots + \rho_k N_k$ where $\rho_i \in \mathbb{R}$ & $N_i \sim N(\mu_i, \sigma_i)$ be a RV. w/ PDF f. This must be an affine combination, that is $\rho_1 + \dots + \rho_k = 1$ & $\rho_i \geq 0$ $\forall i$.

The best fit model is the ρ's & N's that <u>maximizes the total probability of the observed data.</u>

We do this by finding

$$\max_f \prod_{x \in S} f(x) = \max_f \sum_{x \in S} \log f(x).$$

avoids numerical error

Pros: Model gives a lot of information, can assign probabilities to the classification

Cons: Difficult to fit the model,