scalar is its own trace $a = Tr(a)$

Principal component analysis

collection of m points - $\{x^{(1)} \dots x^{(m)}\}$

- Apply lossy compression to points.

These points can be encoded to represent a lower dimensional version of them.

For each point $x^{(i)} \in R^n$ find corresp. code vector $c^{(i)} \in R^l$.

If $l < n$ storing these code points

will occupy lesser memory than original data.

encoding function such that $f(x) = c$

decoding function $\approx x \approx g(f(x))$

PCA can be used as decoding func.

For a simple decoder matrix mul. can be used to map code back to R^n.

$g(c) = Dc$. where $D \in R^{n \times d}$ - matrix defining decoding.

Computing the optimal code for this decoding problem so to make it simpler we constrain the columns of D to be orthogonal to each other.

The columns of D also have to be constrained to have unit norm to avoid learning scaled versions to possibility of many solutions.

To generate optimal code point c^* for each input x, minimize dist between x & reconstruction $g(c^*)$. Use norm to measure distance (refer previous notes for norm). PCA uses L^2 norm.

$$c^* = \arg\min_c \|x - g(c)\|_2$$

squared L^2 norm can be used as both squared L^2 norm & L^2 norm are minimized by same value of c as L^2 norm is non-negative & squaring operation monotonically inc. for non-negative numbers.

$$c^* = \arg\min_c \|x - g(c)\|_2^2$$

function being minimized reduces to

$$(x - g(c))^T (x - g(c))$$

$$= x^T x - x^T g(c) - g(c)^T x + g(c)^T g(c)$$

$$= x^T x - 2 x^T g(c) + g(c)^T g(c)$$

(because scalar $g(c)^T x$ is equal to its transpose).

omit 1st term as it does not depend on c.

$$c^* = \arg\min_c \; -2x^T g(c) + g(c)^T g(c)$$

$$\boxed{g(c) = Dc}$$

$$\therefore c^* = \arg\min_c \; -2x^T Dc + c^T D^T Dc$$

$$= \arg\min_c \; -2x^T Dc + c^T I_1 c$$

$$= \arg\min_c \; -2x^T Dc + c^T c.$$

Applying vector calculus (directional derivatives)

$$\nabla_c(-2x^T Dc + c^T c) = 0$$

$$-2D^T x + 2c = 0$$

$$c = D^T x.$$

From above equation x can be encoded optimally using just a matrix vector operation. To encode a vector : $f_1(x) = D^T x$.

PCA reconstruction : $r(x) = g(f_1(x)) = DD^T x$.

Now to choose encoding matrix D,

lets revisit the idea of minimizing the L^2 distance between inputs & reconstructions

Since same matrix will be used to decode all points, the points can't be considered in isolation. Instead the Frobenius norm of matrix of errors computed for all points must be minimized.

$$D^* = \arg\min_D \sqrt{\sum_{i,j} \left(x_j^{(i)} - r(x^{(i)})_j\right)^2}$$

subject to $D^T D = I_l$.

To find D^*, first case $l = 1$

then D^T is single vector d.

thus $d^* = \arg\min_{d} \sum_{i} \|x^{(i)} - dd^T x^{(i)}\|_2^2$

subject to $\|d\|_2 = 1$.

A scalar is its own transpose

so

$$d^T = \arg\min_{d} \sum_{i} \|x^{(i)} - x^{(i)T} dd\|_2^2$$

$\|d\|_2 = 1$.

$X \in R^{m \times n}$ - matrix formed by stacking all vectors such that $X_{i,:} = x^{(i)T}$.

$$d^* = \arg\min_{d} \|X - Xdd^T\|_F^2, \quad d^T d = 1$$

$$= \arg\min_{d} Tr\left((X - Xdd^T)^T (X - Xdd^T)\right)$$

$$= \arg\min_{d} Tr\left(X^T X - X^T X dd^T - dd^T X^T X + dd^T X^T X dd^T\right)$$

$$= \arg\min_{d} \; Tr(X^T X) - Tr(X^T X d d^T) - Tr(d d^T X^T X) + Tr(d d^T X^T X d d^T).$$

$$= \arg\min_{d} \; -Tr(X^T X d d^T) - Tr(d d^T X^T X) + Tr(d d^T X^T X d d^T)$$

(as terms not having (d) don't affect $\arg\min$)

$$= \arg\min_{d} \; -2\,Tr(X^T X d d^T) + Tr(d d^T X^T X d d^T)$$

(inside a trace order of matrices can be cycled.

$$= \arg\min_{d} \; -2\,Tr(X^T X d d^T) + Tr(X^T X d d^T d d^T),$$

remember the constraint $d d^T = 1$

$$\Rightarrow \arg\min_{d} \; -Tr(X^T X d d^T)$$

$$= \arg\max_{d} \; Tr(X^T X d d^T)$$

$$= \arg\max_{d} \; Tr(d^T X^T X d)$$

Now the above problem can be solved using eigendecomposition (if you can't recognize the pattern see my previous notes). Optimal d is given by eigenvector X^TX corresp. to largest eigenvalue.

Now above case is only for d=1. We can use mathematical induction to generalize.