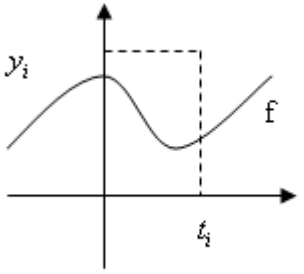
Vandermonde Matrix / Circulant Matrix

P : Given $\left\{ t_i, y_i \middle| i = 1 \sim m \right\}$ find a function f consists with n parameters $\left(m \geq n \right)$ that gives

$$\min_x \sum_{i=1}^{m} \left(y_i - f\left(t_i, x \right) \right)^2$$



$$\Rightarrow \nabla_x \left(\sum_{i=1}^{m} \left(y_i \overset{y_i}{-} f\left(t_i, x \right) \right)^2 \right) = 0$$

$$\Rightarrow \sum_{i=1}^{m} 2\left(y_i - f\left(t_i, x \right) \right)^2 \nabla_x f\left(t_i, x \right) = 0 \ -(VI)$$

Assuming f depends on x linearly, P is called a linear problem.

In this case, we consider

$$f(t,x)=\sum_{j=1}^{n}x_j\phi_j(t)$$

$$(VI)\Rightarrow\sum_{j=1}^{n}\left(y_i-\sum_{j=1}^{n}x_j\phi_j(t_i)\right)\cdot\begin{pmatrix}\phi_1(t_i)\\ \vdots\\ \phi_n(t_i)\end{pmatrix}=0\quad\Rightarrow\begin{cases}\sum_{j=1}^{n}\sum_{i=1}^{m}x_j\phi_j(t_i)\phi_1(t_i)-\sum_{i=1}^{m}y_i\phi_1(t_i)=0\\ \vdots\\ \sum_{j=1}^{n}\sum_{i=1}^{m}x_j\phi_j(t_i)\phi_n(t_i)-\sum_{i=1}^{m}y_i\phi_n(t_i)=0\end{cases}$$

$$\Rightarrow\Phi^{T}_{n\times m}\left[\underset{m\times n}{\Phi}\ \underset{n\times 1}{\vec{x}}-\underset{m\times 1}{\vec{y}}\right]=0\quad----\ (++)$$

$$here\ \vec{x}=\begin{bmatrix}x_1\\ \vdots\\ x_n\end{bmatrix},\ \vec{y}=\begin{bmatrix}y_1\\ \vdots\\ y_n\end{bmatrix},\ \Phi=\begin{bmatrix}\phi_1(t_1) & \cdots & \phi_n(t_1)\\ \phi_1(t_2) & & \phi_n(t_2)\\ \vdots & & \vdots\\ \phi_1(t_m) & \cdots & \phi_n(t_m)\end{bmatrix}_{m\times n}$$

(1) When $\phi_1 = 1,\ \phi_2 = t,\ \phi_3 = t^2,\ \cdots\ \phi_n = t^{n-1}$, one has

$$\Phi = \begin{bmatrix} 1 & t_1 & t_1^2 & \cdots & t_1^{n-1} \\ 1 & t_2 & t_2^2 & \cdots & t_2^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & t_m & t_m^2 & \cdots & t_m^{n-1} \end{bmatrix}$$

The matrix Φ is called the Vandermonde matrix.

It can be shown, Φ is nonsigular when $m = n$.

Solving the equation $\Phi x = y$ is equivalent to find the polynomial $p(t) = \sum_{t=0}^{n-1} x_i t^i$ with degree n-1 such that

$$p(t_1) = y_1 \ \cdots\ p(t_n) = y_n$$

It is well known this polynomial exists and unique.

It is called the Lagrange polynomial.

Now, how to solve $(++)$ efficiently?

Find $p(x)=\sum_{k=0}^{n-1} a_k x^k$ *that satisfy* $p(x_k)=y_k,\ k=0 \sim n\text{-}1$

Consider the Newton representation of the interpolating polynomial p.

$$p(x)=C_0+(x-x_0)\left\{C_1+(x-x_1)\left\{C_2+(x-x_2)\left\{C_3+\cdots\left\{C_{n-2}+(x-x_{n-2})C_{n-1}\right\}\cdots\right\}\right\}\right\}$$

$$=\sum_{k=0}^{n\text{-}1} C_k \prod_{i=0}^{k-1}(x-x_i)$$

$$\Rightarrow C_0 = p(x_0)=y_0$$

$$C_1=\frac{p(x_1)-C_0}{x_1-x_0}=\frac{y_1-y_0}{x_1-x_0}$$

$$C_2=\frac{p(x_2)-C_0-(x_2-x_0)C_1}{x_2-x_1}$$

$$\vdots$$

$$C_{n\text{-}1}=\frac{p(x_{n-1})-\sum_{k=0}^{n-2} C_k \prod_{i=0}^{k-1}(x_{n-1}-x_i)}{\prod_{i=0}^{n-1}(x_{n-1}-x_i)}$$

Algorithm 1

$$\Rightarrow\begin{cases} C(0:n-1)=[y_0,\cdots y_{n-1}] \\ For\ k\ =\ 0:n\text{-}2 \\ \quad For\ i\ =\ n\text{-}1\ to\ k+1 \\ \quad\quad C_i=\dfrac{C_i-C_{i-1}}{x_i-x_{i-k-1}} \\ \quad end \\ end \end{cases}$$

Computing $C_0, C_1 \sim C_{n-1}$ costs $O(n^2)$

Next, we generate $a_k, k = 0 \sim n-1,$ from $C_0, C_1 \sim C_{n-1}$

Consider $p_{n\text{-}1}(x), p_{n-2}(x), \cdots, p_0(x)$ generated by

$$p_{n\text{-}1}(x) = C_{n-1},\ p_k(x) = C_k + (x - x_k)\, p_{k+1},\ k = n\text{-}2 \ \sim\ 0.$$

let $p_k(x) = a_k^{(k)} + a_{k+1}^{(k)} x + \cdots + a_{n-1}^{(k)} x^{n-k-1}$

$$= C_k + (x - x_k)\left[a_{k+1}^{(k+1)} + a_{k+2}^{(k+1)} x + \cdots + a_{n-1}^{(k+1)} x^{n-k-2} \right]$$

$$\Rightarrow a_k^{(k)} = C_k - x_k a_{k+1}^{(k+1)},\ a_{k+1}^{(k)} = a_{k+1}^{(k+1)} - x_k a_{k+2}^{(k+1)},\ a_{k+2}^{(k)} = a_{k+2}^{(k+1)} - x_k a_{k+3}^{(k+1)},\ \cdots$$

$$\Rightarrow (*) \begin{cases} a_k^{(k)} = C_k - x_k a_{k+1}^{(k+1)} \\ For\ i = k+1 :\ n\text{-}2 \\ \qquad a_i^{(k)} = a_i^{(k+1)} - x_k a_{i+1}^{(k+1)} \\ end \\ a_{n-1}^{(k)} = a_{n-1}^{(k+1)} \end{cases}$$

$\Rightarrow$ Excuting (*) from n-2 to 0 gives the coefficients

$$a_0^{(0)},\ a_1^{(0)}, a_2^{(0)}, \cdots, a_{n-1}^{(0)} = a_0, a_1, \cdots a_{n-1}$$

The above process can be written as following:

Algorithm 2:

$$
\left.
\begin{array}{l}
a(0:n\text{-}1) = C(0:n-1) \\
For \;\; k = n\text{-}2 \;\; to \;\; 0 \\
\qquad For \;\; i = k \;\; to \;\; n\text{-}2 \\
\qquad\qquad a_i = a_i - x_k a_{i+1} \\
\qquad end \\
end
\end{array}
\right\} \; operation\ counts \; \mathrm{O}\left(n^2\right)
$$

Remark: (1) Solving the Vandermonde Matrix costs $\mathrm{O}\left(n^2\right)$ operations.

(2) Algorithm 1 and 2 generate accurate solution even when the matrix is ill-conditioned.

(2) Consider $\phi_j(t) = w_{j-1} = \left(e^{-i\frac{2\pi t}{n}} \right)^{j-1}$, $j = 1 \sim n$, where w

is the jth root of $x^n = 1$ $\left(w_j^n = 1 \right)$ and $t_i = i,\ i = 1 \sim n$

$$\Phi_N = \begin{bmatrix} \phi_1(0) & \phi_2(0) & \cdots & \phi_n(0) \\ \phi_1(1) & \phi_2(1) & \cdots & \phi_n(1) \\ \vdots & & & \vdots \\ \phi_1(n-1) & \phi_2(n-1) & \cdots & \phi_n(n-1) \end{bmatrix} \left(\begin{array}{c} let\ w = e^{-i\frac{2\pi}{n}}, w_1 = w,\ w_2 = w^2, \cdots \\ w^{\frac{n}{2}} = -1, \cdots, w_{n-1} = w^{n-1}, w^n = 1 \end{array} \right)$$

$$= \begin{bmatrix} 1 & w_1^0 & w_2^0 & \cdots & w_{n-1}^0 \\ 1 & w_1 & w_2 & \cdots & w_{n-1} \\ 1 & w_1^2 & w_2^2 & \cdots & w_{n-1}^2 \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & w_1^{n-1} & w_2^{n-1} & \cdots & w_{n-1}^{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w & w^2 & \cdots & w^{n-1} \\ 1 & w^2 & w^4 & \cdots & w^{2(n-1)} \\ 1 & w^3 & w^6 & \cdots & w^{3(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & w^{n-2} & w^{2(n-2)} & \cdots & w^{(n-2)(n-1)} \\ 1 & w^{n-1} & w^{2(n-1)} & \cdots & w^{(n-1)^2} \end{bmatrix} \begin{matrix} 0 \\ 1 \\ 2 \\ \vdots \\ \vdots \\ \vdots \\ n-1 \end{matrix}$$

To solve $\Phi_N x = y$ is equivalent to find the coefficients of the fourier expansion of y

(A) Permute Column by [0,2,4,6, … ,N-2,1,3,5, … ,N-1]

$$
\begin{array}{c}
0 \\ 1 \\ \vdots \\ \vdots \\ \frac{N}{2}-1 \\ \hline \frac{N}{2} \\ \frac{N}{2}+1 \\ \vdots \\ \vdots \\ n-1
\end{array}
\left[\begin{array}{ccccc|ccccc}
1 & 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & \cdots & 1 \\
1 & w^2 & w^4 & \cdots & w^{n-2} & w & w^3 & \cdots & \cdots & w^{n-1} \\
1 & \left(w^2\right)^2 & \left(w^4\right)^2 & \cdots & \left(w^{n-2}\right)^2 & w^2 & \left(w^3\right)^2 & \cdots & \cdots & \left(w^{n-1}\right)^2 \\
\vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & & \vdots \\
1 & \left(w^2\right)^{\frac{N}{2}-1} & \left(w^4\right)^{\frac{N}{2}-1} & \cdots & \left(w^{n-2}\right)^{\frac{N}{2}-1} & w^{\frac{N}{2}-1} & \left(w^3\right)^{\frac{N}{2}-1} & \cdots & \cdots & \left(w^{n-1}\right)^{\frac{N}{2}-1} \\ \hline
1 & \left(w^2\right)^{\frac{N}{2}} & \cdots & \cdots & \left(w^{n-2}\right)^{\frac{N}{2}} & w^{\frac{N}{2}} & \left(w^3\right)^{\frac{N}{2}} & \cdots & \cdots & \left(w^{n-1}\right)^{\frac{N}{2}} \\
1 & \left(w^2\right)^{\frac{N}{2}+1} & \cdots & \cdots & \left(w^{n-2}\right)^{\frac{N}{2}+1} & w^{\frac{N}{2}+1} & \left(w^3\right)^{\frac{N}{2}+1} & \cdots & \cdots & \vdots \\
\vdots & \vdots & & & & \vdots & \vdots & & & \vdots \\
\vdots & \vdots & & & & \vdots & \vdots & & & \vdots \\
1 & \left(w^2\right)^{n-1} & \cdots & \cdots & \left(w^{n-2}\right)^{n-1} & w^{n-1} & \left(w^3\right)^{n-1} & \cdots & \cdots & \left(w^{n-1}\right)^{n-1}
\end{array}\right]
$$

$$
\overset{w^n=1}{=}
\left[\begin{array}{ccccc|ccccc}
1 & 1 & \cdots & \cdots & 1 & 1 & 1 & \cdots & \cdots & 1 \\
w^2 & w^4 & \cdots & \cdots & w^{n-2} & w & w^3 & \cdots & \cdots & w^{n-1} \\
w^4 & w^8 & \cdots & \cdots & w^{n-4} & w^2 & w^6 & \cdots & \cdots & w^{n-2} \\
\vdots & \vdots & & & \vdots & \vdots & \vdots & & & \vdots \\
w^{n-2} & w^{n-4} & \cdots & \cdots & w^2 & w^{\frac{N}{2}-1} & w^{\frac{N}{2}-3} & \cdots & \cdots & w^{\frac{N}{2}+1} \\ \hline
1 & 1 & \cdots & \cdots & 1 & -1 & -1 & \cdots & \cdots & -1 \\
w^2 & w^4 & \cdots & \cdots & w^{n-2} & -w & \cdots & & & -w^{n-1} \\
w^4 & w^8 & \cdots & \cdots & w^{n-4} & \vdots & & & & \vdots \\
\vdots & \vdots & & & \vdots & \vdots & & & & \vdots \\
w^{n-2} & w^{n-4} & \cdots & & w^2 & -w^{\frac{N}{2}-1} & \cdots & & & -w^{\frac{N}{2}+1}
\end{array}\right]
\quad ---(**)
$$

$$let\ \Omega_{\frac{N}{2}} = \begin{bmatrix} 1 & & & \\ & w & & 0 \\ & & w^2 & \\ & 0 & & \ddots \\ & & & & w^{\frac{N}{2}-1} \end{bmatrix},\ \Phi_{\frac{N}{2}} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ w^2 & w^4 & \cdots & w^{n-2} \\ w^4 & w^8 & \cdots & w^{n-4} \\ \vdots & \vdots & & \\ w^{n-2} & w^{n-4} & \cdots & w^2 \end{bmatrix}$$

$$(**) \underset{(B)}{=} \left[\begin{array}{c|c} \Phi_{\frac{N}{2}} & \Omega_{\frac{N}{2}}\Phi_{\frac{N}{2}} \\ \hline \Phi_{\frac{N}{2}} & -\Omega_{\frac{N}{2}}\Phi_{\frac{N}{2}} \end{array}\right] \underset{(A)\&(B)}{\overset{repeat}{\Rightarrow}} \Phi_{\frac{N}{2}} = \left[\begin{array}{c|c} \Phi_{\frac{N}{4}} & \Omega_{\frac{N}{4}}\Phi_{\frac{N}{4}} \\ \hline \Phi_{\frac{N}{4}} & -\Omega_{\frac{N}{4}}\Phi_{\frac{N}{4}} \end{array}\right],\ \Omega_{\frac{N}{4}} = \begin{bmatrix} 1 & & & \\ & w^2 & & 0 \\ & & w^4 & \\ & 0 & & \ddots \\ & & & & w^{\frac{N}{2}-2} \end{bmatrix}$$

$$\Rightarrow \text{keep doing until } \dim(\Phi) = 1.$$

The above process is the matrix version of the well known FFT (Fast Fourier Transformation) procedure.

exercise:

Show that (1) $\frac{1}{N}\Phi^{*}\Phi = I$ (2) Φ is nonsingular.

To solve $\Phi^{*}\left(\Phi x - y\right) = 0 \equiv$ To solve $\Phi x = y$

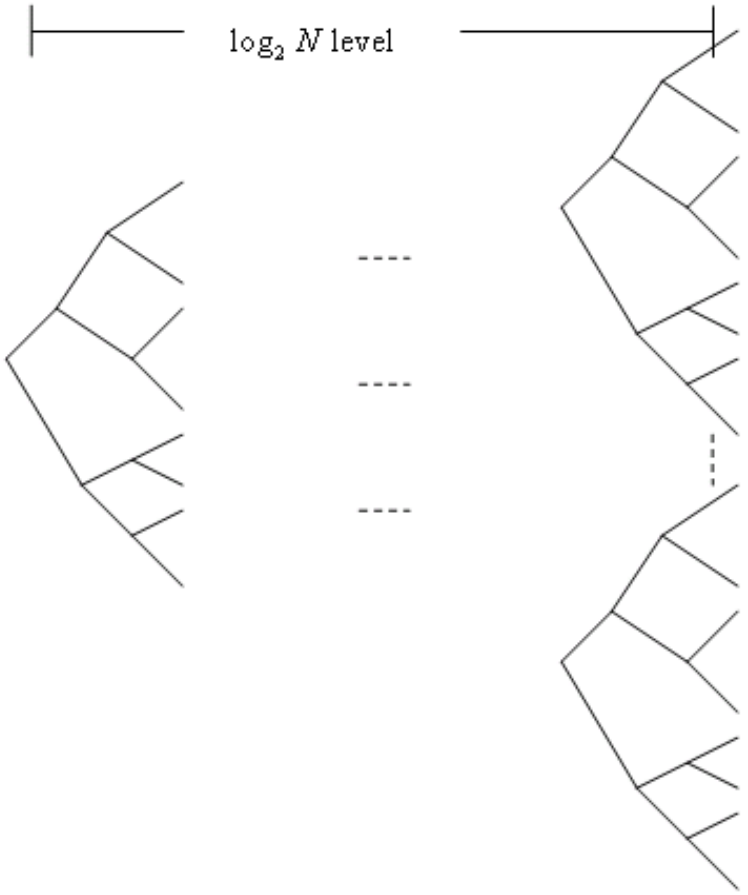
We only needs to apply $\frac{1}{N}\Phi^{*}$ to the equation.

Moreover, $x = \frac{1}{N}\Phi^{*} y$ can be computed extremely fast due to the above recursive relation.

Remark: In fact, the algorithm can be implemented in $\mathrm{O}\left(n \log n\right)$ operations.

By, $\Phi_k x^{(k)} = \left[\dfrac{\Phi_{\frac{k}{2}} x_{even}^{(k)} + \Omega_{\frac{k}{2}} \Phi_{\frac{k}{2}} x_{odd}^{(k)}}{\Phi_{\frac{k}{2}} x_{even}^{(k)} - \Omega_{\frac{k}{2}} \Phi_{\frac{k}{2}} x_{odd}^{(k)}} \right]$,

$\Phi_N x$ can now be computed recursively starting from computing $\Phi_1 x_i, i = 0 \cdots n-1$.



Total cost for FFT

$$= 3 \cdot 2^{n-1} \cdot \frac{N}{2^{N-1}} + \cdots\cdots + \left(3 \cdot 2^2 \frac{N}{2^2} \right) (3 \cdot 2) \frac{N}{2} + N$$

$$< 3N \log_2 N$$

Circulant Matirx :

$$C(v)=\begin{bmatrix} v_0 & v_{n-1} & v_{n-2} & \cdots & v_1 \\ v_1 & v_0 & v_{n-1} & \cdots & v_2 \\ \vdots & & \ddots & \ddots & \vdots \\ v_{n-2} & \cdots & \cdots & v_0 & v_{n-1} \\ v_{n-1} & \cdots & \cdots & v_1 & v_0 \end{bmatrix}$$ is called circulant matirx.

Given a vertor $x \in \mathbb{R}^n$, the product $C(v) \cdot x$ is called the convolution of x with v and the product of $C(v)^{-1} \cdot y$ is called the deconvolution of y from v.

The matrix is formed by $C(v)=\left[v,\ sv,\ s^2v,\ \cdots,\ s^{n-1}v\right]$ where

$$S=\begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & \cdots & 0 \\ & 1 & \ddots & 0 & \vdots \\ & & \ddots & \ddots & \vdots \\ 0 & & & 1 & 0 \end{bmatrix}$$ is called a "downshifted " matrix

First let's show charateristic polynomial of $C(v)$ has n roots.

Prove that the characteristic root of the circulant $C(v)$ is

$$y = v_1 r^{n-1} + v_2 r^{n-2} + \cdots + v_{n-1} r + v_0 \quad \text{where } r^n = 1$$

let $y_1 = v_1 r^{n-1} + v_2 r^{n-2} + \cdots + v_{n-1} r + v_0$.

We have $r y_1 = v_2 r^{n-1} + \cdots + v_{n-1} r^2 + v_o r + v_1$

$$r^2 y_1 = v_3 r^{n-1} + \cdots + v_{n-1} r^3 + v_0 r^2 + v_1 r + v_2$$

$$\vdots$$

$$r^{n-1} y_1 = v_0 r^{n-1} + v_1 r^{n-2} + \cdots + v_{n-2} r + v_{n-1}$$

$$\Rightarrow \begin{bmatrix} v_0 & v_{n-1} & \cdots & v_2 & v_1 \\ v_1 & v_0 & \cdots & v_3 & v_2 \\ & \ddots & \ddots & \vdots & \vdots \\ & & \ddots & v_0 & v_{n-1} \\ v_{n-1} & v_{n-2} & \cdots & v_1 & v_0 \end{bmatrix} \begin{bmatrix} 1 \\ r \\ r^2 \\ \vdots \\ r^{n-1} \end{bmatrix} = y_1 \begin{bmatrix} 1 \\ r \\ r^2 \\ \vdots \\ r^{n-1} \end{bmatrix}$$

$\Rightarrow y_1$ is an e-value of the circulant matrix.

Next, consider $\Phi(j,k) = \left(w_{j-1}^{k-1}\right) = e^{-\frac{2\pi i}{n}(j-1)(k-1)}$, $\Phi^{-1} = \frac{1}{N}\Phi^{*}$ and

$\lambda(j) = \sum_{e=1}^{n} v(l) e^{-\frac{2\pi i}{n}(j-1)(l-1)}$.

By direct computing, it can be shown C is diagonalizable and

$$C = \Phi^{-1}\Lambda\Phi \quad \text{where } \Lambda = \begin{bmatrix} \lambda(1) & & 0 \\ & \ddots & \\ 0 & & \lambda(n) \end{bmatrix}.$$

Moreover, $\Lambda = diag(\lambda_i) = diag(\Phi \cdot v)$, here $v = [v_0, v_1, \cdots, v_{n-1}]$.

The above equalities implies that the convolution $y = Cx$

can be carried out by 3 FFT and one vector pointwise multiplication.

The convolution algorithm is as following:

$$\begin{cases} 1.\ \tilde{x} = \Phi x \\ 2.\ \tilde{v} = \Phi v \\ 3.\ z = \tilde{v} * \tilde{x}\ \text{(pointwise multiplication)} \\ 4.\ y = \Phi z \end{cases}$$

Similarly, the deconvolution can be done in the same cost!

$\left(\text{The cost is down to } n\log n \text{ from the cost } \mathrm{O}\left(n^2\right) \text{ of the direct matrix vector product}\right)$

Now, let's come back to solve the system (++) for the case $m \gg n$

$$\Phi^{T}_{n\times m}\left[\Phi_{m\times n}x - y\right] = 0 \quad --- \quad (++)$$

Suppose one can find some transformation Q^T such that

$$Q^T\Phi_{m\times n} = \begin{bmatrix}\widetilde{R}_{n\times n}\\ \hline 0\end{bmatrix} = R \text{ here } \widetilde{R} \text{ is nonsingular.} \left(\begin{array}{l} Q^T \text{ exists since} \\ \text{we can always use } G.E. \\ \text{to do it.}\end{array}\right)$$

Then (++) $\Rightarrow$ $R^T Q^T\left[QRx - y\right] = 0$

$$\Rightarrow R^T\left[\left(Q^T Q\right)Rx - Q^T y\right] = 0$$

Suppose one can find Q that is orthogonal.$\left(Q^T Q = I\right)$

Then the system becomes

$$R^T\left[Rx - Q^T y\right] = 0 \quad \left(\left[\widetilde{R}^* \middle| 0\right]\left(\left[\frac{\widetilde{R}}{0}\right]x - Q^T y\right) = 0\right)$$

One only needs to solve $\widetilde{R}_{n\times n}\,\tilde{x} = \tilde{y}$.

This is easy to solve since $n \ll m$.

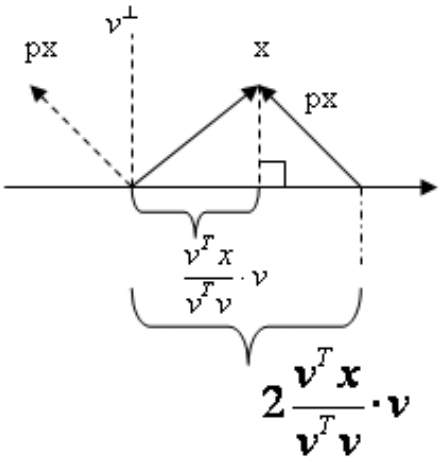
So the question now becomes that given a matrix $A_{m\times n}$, how to find an orthogonal transformation Q such that $A = QR$

We now introduce the *QR* decomposition of *A* in the following:

Consider the House-Holder transformation matrix

$$p = I - \frac{2}{v^T v} v v^T$$

$$\Rightarrow px = x - \frac{2}{v^T v} v\left(v^T x\right) = x - \frac{2\left(v^T x\right)}{v^T v} \cdot v$$



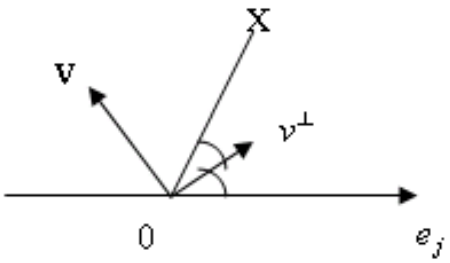
It is clear that *px* is the reflection of *x* with respect to $v^{\perp}$

Basic idea:

Given a vector x, try to find v such that

(&) $px = \beta\vec{e}_1$ $\left(\text{or } \vec{e}_j \text{ for any given } j\right)$,

here β is a constant.



$$\text{consider } \left.\begin{aligned} v &= x + \alpha e_1 \\ v^T x &= x^T x + \alpha x_1 \\ v^T v &= x^T x + 2\alpha x_1 + \alpha^2 \end{aligned}\right\} \Rightarrow px = x - \frac{2vv^T}{v^T v}x = x - \frac{2\left(x+\alpha e_1\right)\cdot\left(x^T x + \alpha x_1\right)}{x^T x + 2\alpha x_1 + \alpha^2}$$

$$= \left(1 - \frac{2\left(x^T x + \alpha x_1\right)}{\|x\|^2 + 2\alpha x_1 + \alpha^2}\right)x - \underbrace{\frac{2\alpha\left(x^T x + \alpha x_1\right)}{\|x\|^2 + 2\alpha x_1 + \alpha^2}}_{\beta} e_1$$

To achieve (&), we simply coose α such that $1 - \dfrac{2\left(x^T x + \alpha x_1\right)}{\|x\|^2 + 2\alpha x_1 + \alpha^2} = 0 \Rightarrow \begin{matrix} \alpha = \|x\| \\ \beta = \alpha = \|x\| \end{matrix}$

Notice that

(1) $p^T p = I \Rightarrow p$ is an orthogonal transformation.

(2) Given a matrix $A = [A_1, A_2, \cdots A_n]$, one can construct

p_1 such that $p_1 A_1 = \beta_1 e_1$

$$p_1 A = \begin{bmatrix} \beta_1 & * & * & \cdots & * \\ 0 & & * & \cdots & * \\ 0 & \tilde{A}_2 & * & & * \\ \vdots & & \vdots & & \vdots \\ 0 & & * & \cdots & * \end{bmatrix}$$

$$\| \; x$$

Similarly, by constructions p_2 such that $p_2 \tilde{A}_2 = \begin{bmatrix} \beta_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$, we have $\left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & p_2 \end{array}\right] p_1 A = \begin{bmatrix} \beta_1 & * & * & * & \cdots & * \\ 0 & \beta_2 & * & * & \cdots & * \\ 0 & 0 & & * & \cdots & * \\ \vdots & \vdots & \widetilde{A}_3 & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & & \vdots \\ 0 & 0 & & * & \cdots & * \end{bmatrix}$

Repeating the above process, one has

$$Q^T = \underbrace{\left[\begin{array}{c|c} I & 0 \\ \hline 0 & p_n \end{array}\right]}_{\mathrm{P}_n} \cdot \underbrace{\left[\begin{array}{c|c} I & 0 \\ \hline 0 & p_{n-1} \end{array}\right]}_{\mathrm{P}_{n-1}} \cdots \underbrace{\left[\begin{array}{c|c} I & 0 \\ \hline 0 & p_2 \end{array}\right]}_{\mathrm{P}_2} \cdot \underbrace{p_1}_{\mathrm{P}_1}$$ is orthogonal.

and $Q^T A = R$ where $R = \left[\begin{array}{ccc} \beta_1 & & * \\ & \beta_2 & \\ & & \ddots \\ & & \ddots \\ \hline & 0 & \end{array}\right]$

In practice, we never construct $\mathrm{P}_i,\ i = 1 \sim n,$ explicitly.

Since $pA = \left(I - \dfrac{2vv^T}{v^T v} \right) A \underset{r=\frac{2}{v^T v}}{=} \left(I - rvv^T \right) A = \underbrace{A - rvw^T}_{\text{cost } \mathrm{O}(m \times n)},\ \underbrace{w = vA^T v}_{m \times n \text{ operation}}$

Total cost for QR is about $\sum_{i=0}^{n\text{-}1} (m - i)(n - i) \approx \mathrm{O}\left(mn^2\right)$