Markov Chain and Matrices Methods

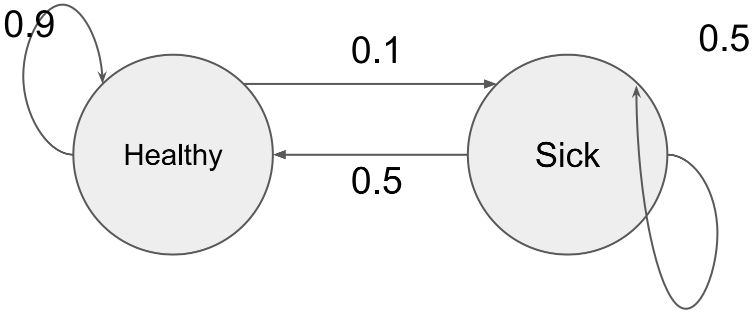
Diego Useche - dh.useche@uniandes.edu.co

Metodos Computacionales I

Physics Department, Universidad de los Andes, Bogotá

Markov Chain

$$A\begin{bmatrix}0.9 & 0.5\\0.1 & 0.5\end{bmatrix}\begin{bmatrix}1\\ \end{bmatrix}$$



It can start in any state

T	Healthy	Sick
0	1	0
1	0.9	0.1
2	(0.9*0.9) + (0.1*0.5) = 0.86	(0.9*0.1) + (0.1*0.5) = 0.14
3	(0.86*0.9) + (0.14*0.5) = 0.844	(0.86*0.1) + (0.14*0.5) = 0.156
	...	...
t →∞	0.833	0.167

Estable.

$$\begin{bmatrix} 0.9 & 0,5 \\ 0.1 & 0,5 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} H \\ S \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

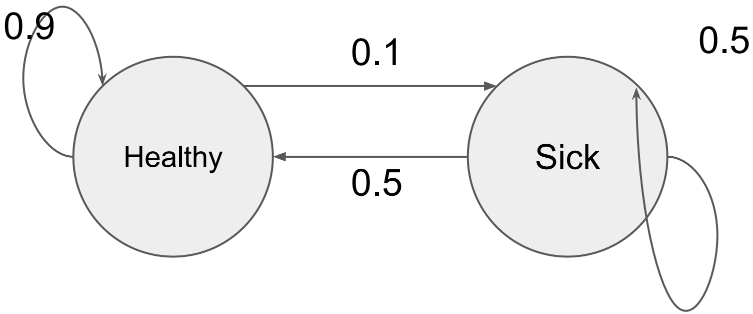
$$= \begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix} = \begin{bmatrix} H_1 \\ S_1 \end{bmatrix}$$

$$\begin{bmatrix} H \\ S \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0.9 & 0,5 \\ 0.1 & 0,5 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

Stationary Distribution

$$\vec{v}_{t+1} = A\vec{v}_t$$



T	Healthy	Sick
t →∞	0.833	0.167

$$\vec{v}_1 = A\vec{v}_0$$
$$\vec{v}_2 = A\vec{v}_1$$

$$\vec{v}_n = \underbrace{AAA\cdots A}_{n}\vec{v}_0$$

$$\vec{v}^* = A\vec{v}_*$$
$$= A\begin{bmatrix}0.833\\0.167\end{bmatrix} = \begin{bmatrix}0.833\\0.167\end{bmatrix}$$

Valores Propios y Vectores Propios

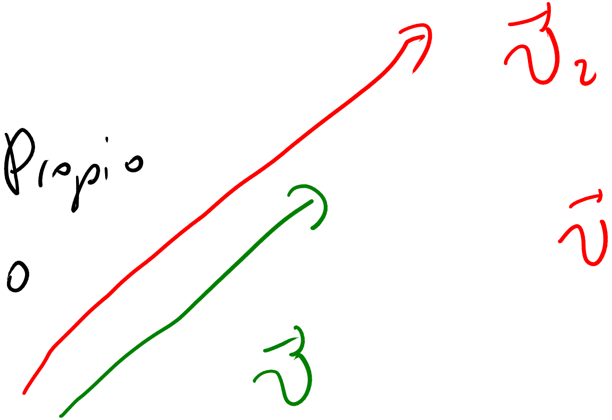
$$A\vec{v} = \lambda \vec{v}$$

$\vec{v}^* = A\vec{v}^*$

(1) $\vec{v}^* = A\vec{v}^*$

$\vec{v}^*$ Vector Propio

1 Valor Propio

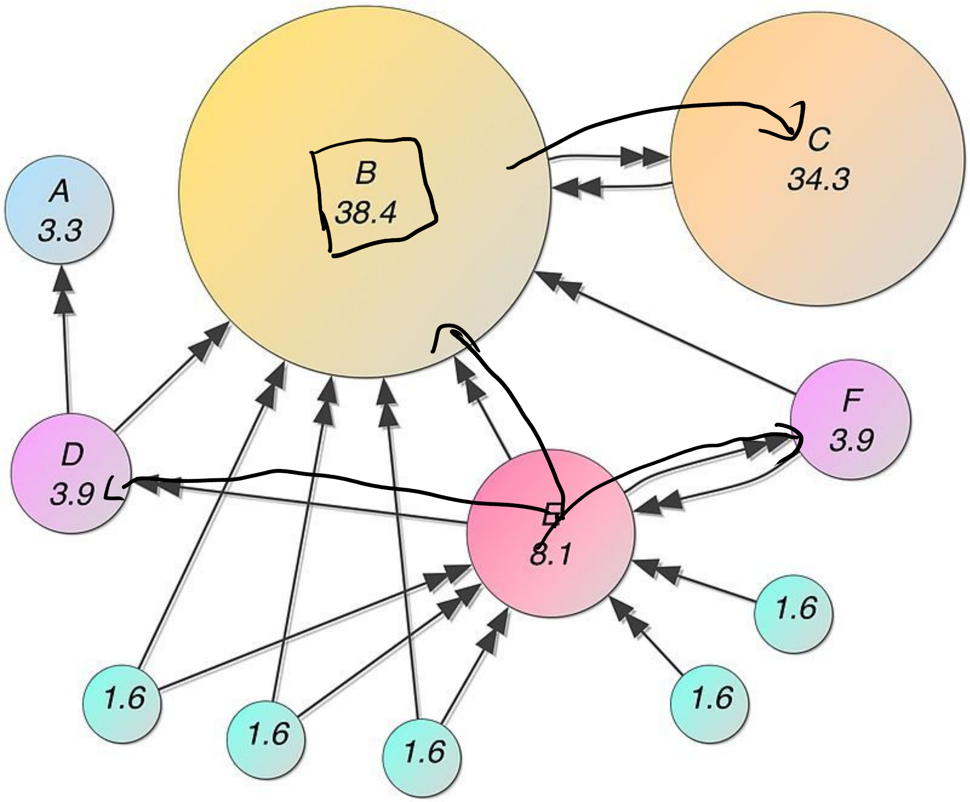


$\vec{v}_2 = \lambda \vec{v}$

$\vec{v}_2 = A\vec{v}$

Page Rank

- Similar to a Markov Chain to determine the connected web pages.
- Built as product of a PhD research by Sergei Brin and Larry Page.
- Used to improve web searching.
- The origin of google.



Gaussian Elimination Algorithm

$$\begin{bmatrix} 1 & 2 & -1 & 1 \\ 2 & 7 & 4 & 2 \\ -1 & 4 & 13 & -1 \end{bmatrix} \to \begin{bmatrix} 1 & 2 & -1 & 1 \\ 0 & 3 & 6 & 0 \\ -1 & 4 & 13 & -1 \end{bmatrix}$$

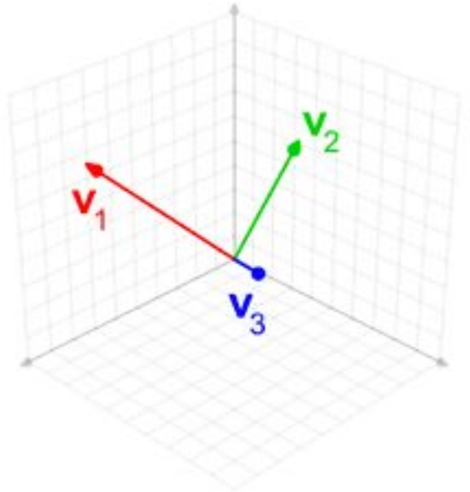
$$\to \begin{bmatrix} 1 & 2 & -1 & 1 \\ 0 & 3 & 6 & 0 \\ 0 & 6 & 12 & 0 \end{bmatrix} \to \begin{bmatrix} 1 & 2 & -1 & 1 \\ 0 & 3 & 6 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

QR Decomposition

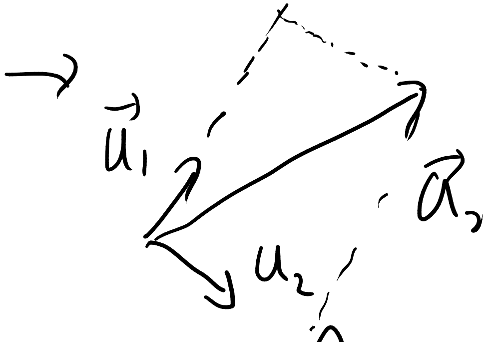
- Let A and mxn matrix
- Then $A = QR$
  - Where Q is an ortogonal matrix (the rows are all orthonormal) , R is an upper triangular matrix.
- Algorithm
  - Gram–Schmidt Algorithm
- Applications:
  - Least Squares

Gram-Schmidt Algorithm

- Let A be a matrix
- Normalize the first row vector , that is our first orthonormal vector(v1)
- Project the second row vector of A over v1, find the span of the projection and normalize (v2), Etc
- Save the coordinates of the original vectors, on the orthonormal basis, in the R matrix.



$$A = \begin{bmatrix} | & | & & | \\ \vec{a}_1 & \vec{a}_2 & \cdots & \vec{a}_n \\ | & | & & | \end{bmatrix}$$



$$\vec{v}_1 = \frac{\vec{a}_1}{\|\vec{a}_1\|}$$

$$\mathrm{Proj}_{\vec{u}_1} \vec{a}_2 = \hat{u}_1 \cdot \vec{a}_2$$

$$\vec{a}_1 = R_{11} \hat{u}_1$$

$$\vec{a}_2 = R_{12} \hat{u}_1 + R_{12} \hat{u}_2$$

$$\vec{a}_3 = R_{13} \hat{u}_1 + R_{23} \hat{u}_2 + R_{33} \hat{u}_3$$

$$\vec{a}_2 = |a_2^{\parallel}| \hat{u}_1 + |a_2^{\perp}| \hat{u}_2$$

$$\begin{bmatrix} | & | & & | \\ \vec{a}_1 & \vec{a}_2 & \cdots & a_n \\ | & | & & | \end{bmatrix} = \begin{bmatrix} \hat{u}_1 & \hat{u}_2 & \cdots \end{bmatrix} \begin{bmatrix} R_{11} & R \end{bmatrix}$$

$$\begin{bmatrix} | & | & | \\ \vec{a}_1 & \vec{a}_2 & \vec{a}_3 \\ | & | & | \end{bmatrix} = \begin{bmatrix} | & | & | \\ \vec{u}_1 & \vec{u}_2 & \vec{u}_3 \\ | & | & | \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ 0 & R_{22} & R_{23} \\ 0 & 0 & R_{33} \end{bmatrix}$$

$$\begin{bmatrix} | \\ \vec{a}_1 \\ | \end{bmatrix} = R_{11} \begin{bmatrix} | \\ \vec{u}_1 \\ | \end{bmatrix} \qquad \begin{bmatrix} | \\ \vec{a}_2 \\ | \end{bmatrix} = R_{12} \begin{bmatrix} | \\ \vec{u}_1 \\ | \end{bmatrix} + R_{22} \begin{bmatrix} | \\ \vec{u}_2 \\ | \end{bmatrix}$$

QR Decomposition

$$\vec{u}_i \cdot \vec{u}_j = \delta_{ij}$$

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

$$Q = \begin{array}{c} \begin{array}{ccc} \vec{u}_1 & \vec{u}_2 & \vec{u}_3 \end{array} \\ \begin{bmatrix} 1/2 & 1/\sqrt{2} & 0 \\ 1/2 & 0 & -1/\sqrt{2} \\ 1/2 & -1/\sqrt{2} & 0 \\ 1/2 & 0 & 1/\sqrt{2} \end{bmatrix} \end{array}$$

$$R = \begin{bmatrix} 2 & 2 & 4 \\ 0 & 2\sqrt{2} & -\sqrt{2} \\ 0 & 0 & \sqrt{2} \end{bmatrix}$$

Spectral Decomposition

- Let A be a symmetric matrix ($A^T = A$)
- Then $A = V\Delta V^T$

  eigenvectores

  eigenvalores

  - Where Δ is a diagonal matrix, and V is an orthonormal matrix.
  - Δ is formed by the eigenvalues of A, V by the eigenvectors
- Algorithm
  - QR Algorithm, divide-and-conquer, Jacobi Algorithm , $O(n^3)$
- Applications:
  - Find the eigenstate of a system.

Spectral Decomposition $\quad A = UDU.T$

$$A = \begin{bmatrix} 9 & 3 & 9 \\ 3 & 17 & -3 \\ 9 & -3 & 9 \end{bmatrix};$$

$$A = U \Lambda U^T$$

$$\begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix} = \begin{array}{c} \begin{matrix} \vec{v}_1 \qquad & \vec{v}_2 \qquad & \vec{v}_3 \end{matrix} \\ \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{38}} & \frac{3}{\sqrt{19}} \\ 0 & -\frac{6}{\sqrt{38}} & -\frac{1}{\sqrt{19}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{38}} & -\frac{3}{\sqrt{19}} \end{bmatrix} \end{array};$$

$$D = \begin{bmatrix} 18 & 0 & 0 \\ 0 & 18 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

λ_1 λ_2 λ_3

https://www.chegg.com/homework-help/

QR Algorithm

Find a series of unitary operations

$$(U_k^* \cdots U_1^*)A(U_1 \cdots U_k) \to T \text{ (triangular) as } k \to \infty.$$

$$A = \begin{matrix} 1 & 1 & 1 & 1 & 1 \\ 16 & 8 & 4 & 2 & 1 \\ 81 & 27 & 9 & 3 & 1 \\ 256 & 64 & 16 & 4 & 1 \\ 625 & 125 & 25 & 5 & 1 \end{matrix}$$

```
>> for k=1:20, [Q,R]=qr(A); A=Q'*A*Q;  end
>> A
```

$$\begin{matrix} 45.9604 & \lambda_1 \\ -28.9437 & \\ 6.7954 & \\ -0.8496 & \\ 0.0375 & \lambda_5 \end{matrix}$$

$$A = \begin{matrix} 45.9697 & 333.0696 & -514.7544 & 307.7187 & -66.7932 \\ -0.0021 & -28.9530 & 62.9322 & -47.4590 & 12.6896 \\ -0.0000 & -0.0000 & 6.7954 & -9.3745 & 3.5876 \\ -0.0000 & -0.0000 & 0.0000 & -0.8496 & 0.6473 \\ -0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0375 \end{matrix}$$

$$
\begin{cases}
A_0 = Q_0 R_0 \\
A_1 = Q_0^T A_0 Q_0 \\
A_1 = Q_1 R_1 \\
A_2 = Q_1^T A_1 Q_1 \\
\vdots
\end{cases}
$$

(1) $R^* = Q_n^T \cdots Q_1^T Q_0^T A_0 Q_0 Q_1 \cdots Q_n$

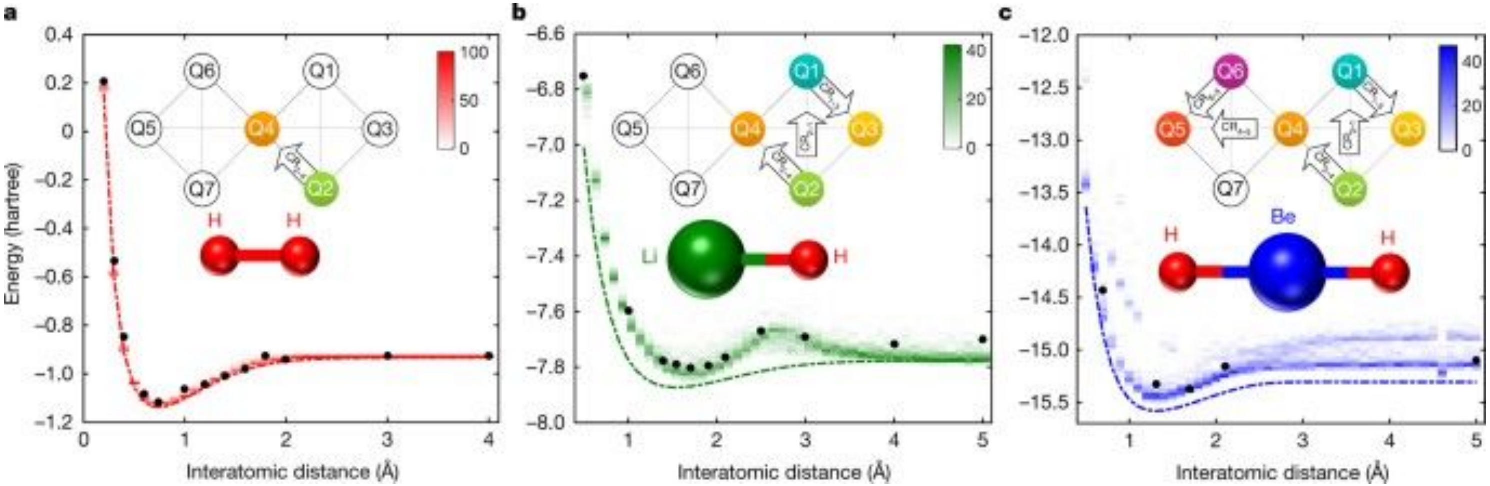
(2)

(3) $R^* = \begin{bmatrix} \lambda_1 & 0 & \cdots & \cdots \\ & \lambda_2 & \cdots & \cdots \\ & & \ddots & \vdots \\ & & & \lambda_n \end{bmatrix}$

Solving Ground States of Molecules with VQE

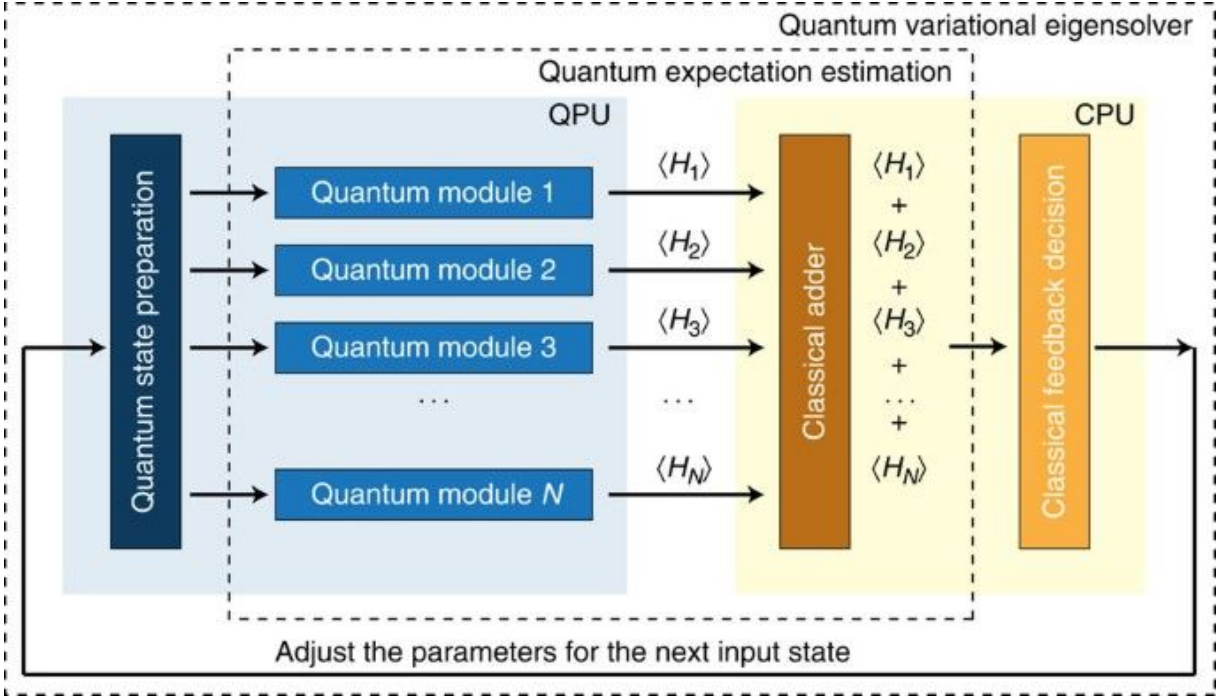
Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets

Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M. Chow & Jay M. Gambetta



A variational eigenvalue solver on a photonic quantum processor

Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik & Jeremy L. O'Brien



References

G. W. Stewart, "The decompositional approach to matrix computation," in *Computing in Science & Engineering*, vol. 2, no. 1, pp. 50-59, Jan.-Feb. 2000. doi: 10.1109/5992.814658

T. Shifrin M.Adams, “Linear Algebra: A geometric approach” Second Edition 2011

https://www.youtube.com/watch?v=_neGVEBjLJA

http://people.duke.edu/~kh269/teaching/notes/MetropolisExplanation.pdf

http://www.mit.edu/~ilkery/papers/MetropolisHastingsSampling.pdf

References

https://www.csie.ntu.edu.tw/~lyuu/theses/thesis_r95922165.pdf

https://blog.stata.com/2016/11/15/introduction-to-bayesian-statistics-part-2-mcmc-and-the-metropolis-hastings-algorithm/

https://www.coursera.org/lecture/bayesian-methods-in-machine-learning

Kandala, A., Mezzacapo, A., Temme, K. *et al.* Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* 549, 242–246 (2017). https://doi.org/10.1038/nature23879

Peruzzo, A., McClean, J., Shadbolt, P. *et al.* A variational eigenvalue solver on a photonic quantum processor. *Nat Commun* 5, 4213 (2014). https://doi.org/10.1038/ncomms5213